CODING AND ROBOTICS

National Curriculum Statement (NCS)

Curriculum and Assessment Policy Statement

SENIOR PHASE
GRADE 7 – 9

basic education
Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

Read to Lead
A Reading Nation is a Leading Nation

2030 NDP

30 YEARS OF FREEDOM

Department of Basic Education

222 Struben Street

Private Bag X895

Pretoria 0001

South Africa

Tel: +27 12 357 3000

Fax: +27 12 323 0601

120 Plein Street

Private Bag X9023

Cape Town 8000

South Africa

Tel: +27 21 465 1701

Fax: +27 21 461 8110

Website: http://www.education.gov.za

ISBN: 978-1-4315-3991-8

Design and Layout by: Department of Basic Education

CURRICULUM AND ASSESSMENT POLICY STATEMENT (CAPS)

CODING AND ROBOTICS

SENIOR PHASE
GRADE 7 – 9

FOREWORD BY THE MINISTER

Our national curriculum is the culmination of our efforts over a period of seventeen years to transform the curriculum bequeathed to us by apartheid. From the start of democracy, we have built our curriculum on the values that inspired our Constitution (Act 108 of 1996). The Preamble to the Constitution states that the aims of the Constitution are to:

- heal the divisions of the past and establish a society based on democratic
- values, social justice and fundamental human rights;
- improve the quality of life of all citizens and free the potential of each person;
- lay the foundations for a democratic and open society in which government is based on the will of the people and every citizen is equally protected by law; and
- build a united and democratic South Africa able to take its rightful place as a sovereign state in the family of nations.

Education and the curriculum have an important role to play in realising these aims. In 1997 we introduced outcomes-based education to overcome the curricular divisions of the past, but the experience of implementation prompted a review in 2000. This led to the first curriculum revision: the Revised National Curriculum Statement Grades R-9 and the National Curriculum Statement Grades 10-12 (2002).

Ongoing implementation challenges resulted in another review in 2009 and we revised the Revised National Curriculum Statement (2002) and the National Curriculum Statement Grades 10-12 to produce this document.

From 2012 the two National Curriculum Statements, for Grades R-9 and Grades 10-12 respectively, are combined in a single document and will simply be known as the National Curriculum Statement Grades R-12. The National Curriculum Statement for Grades R-12 builds on the previous curriculum but also updates it and aims to provide clearer specification of what is to be taught and learnt on a term-by-term basis.

The National Curriculum Statement Grades R-12 represents a policy statement for learning and teaching in South African schools and comprises of the following:

(a) Curriculum and Assessment Policy Statements (CAPS) for all approved subjects listed in this document;

(b) National policy pertaining to the programme and promotion requirements of the National Curriculum Statement Grades R-12; and

(c) National Protocol for Assessment Grades R-12.

MRS ANGIE MOTSHEKGA, MP

MINISTER OF BASIC EDUCATION

CONTENTS

TABLES AND FIGURES

Tables

Figures

1 SECTION 1
INTRODUCTION TO THE CURRICULUM AND ASSESSMENT POLICY STATEMENT FOR CODING AND ROBOTICS INTERMEDIATE PHASE (GRADE 4 – 6)

1.1 BACKGROUND

The *National Curriculum Statement Grades R – 12 (NCS)* stipulates policy on curriculum and assessment in the schooling sector.

To improve implementation, the National Curriculum Statement was amended, with the amendments coming into effect in January 2012. A single comprehensive Curriculum and Assessment Policy document was developed for each subject to replace Subject Statements, Learning Programme Guidelines and Subject Assessment Guidelines in Grades R - 12.

1.2 OVERVIEW

(a) The *National Curriculum Statement Grades R – 12 (January 2012)* represents a policy statement for learning and teaching in South African schools and comprises the following:

- (i) National Curriculum and Assessment Policy Statements for each approved school subject*;*
- (ii) The policy document, National policy pertaining to the programme and promotion requirements of the National Curriculum Statement Grades R – 12; and
- (iii) The policy document, National Protocol for Assessment Grades R – 12 (January 2012).

(b) The *National Curriculum Statement Grades R – 12 (January 2012)* replaces the two current national curricula statements, namely the

- *(i) Revised National Curriculum Statement Grades R - 9, Government Gazette No. 23406* of 31 May 2002*,* and
- (ii) *National Curriculum Statement Grades 10 - 12 Government Gazettes, No. 25545 of 6 October 2003* and *No. 27594* of 17 May 2005.

(c) The national curriculum statements contemplated in subparagraphs (a) and (b) comprise the following policy documents which will be incrementally repealed by the *National Curriculum Statement Grades R – 12 (January 2012)* during the period 2012-2014:

- (i) The Learning Area/Subject Statements, Learning Programme Guidelines and Subject Assessment Guidelines for Grades R - 9 and Grades 10 – 12;
- (ii) The policy document, *National Policy on assessment and qualifications for schools in the General Education and Training Band d*, promulgated in *Government Notice No. 124* in *Government Gazette No. 29626* of 12 February 2007;
- (iii) The policy document, the *National Senior Certificate: A qualification at Level 4 on the National Qualifications Framework (NQF)*, promulgated in *Government Gazette No.27819* of 20 July 2005;
- (iv) The policy document, *An addendum to the policy document, the National Senior Certificate: A qualification at Level 4 on the National Qualifications Framework (NQF), regarding learners with special needs***,** published in *Government Gazette, No.29466* of 11 December 2006, is incorporated in the policy document, *National policy pertaining to the programme and promotion requirements of the National Curriculum Statement Grades R – 12; and*
- (v) The policy document, *An addendum to the policy document, the National Senior Certificate: A qualification at Level 4 on the National Qualifications Framework (NQF), regarding the National Protocol for Assessment (Grades R – 12)*, promulgated in *Government Notice No.1267* in *Government Gazette No. 29467* of 11 December 2006.

(c) The policy document, *National policy pertaining to the programme and promotion requirements of the National Curriculum Statement Grades R – 12*, and the sections on the Curriculum and Assessment Policy as contemplated in Chapters 2, 3 and 4 of this document, constitute the norms and standards of the *National Curriculum Statement Grades R – 12.* It will therefore, in terms of *section 6A* of the *South African*

Schools Act, 1996 (Act No. 84 of 1996,) form the basis for the Minister of Basic Education to determine minimum outcomes and standards, as well as the processes and procedures for the assessment of learner achievement to be applicable to public and independent schools.

1.3 GENERAL AIMS OF THE SOUTH AFRICAN CURRICULUM

- The *National Curriculum Statement Grades R - 12* gives expression to the knowledge, skills and values worth learning in South African schools. This curriculum aims to ensure that children acquire and apply knowledge and skills in ways that are meaningful to their own lives. In this regard, the curriculum promotes knowledge in local contexts, while being sensitive to global imperatives.
- The National Curriculum Statement Grades R - 12 serves the purposes of:
    - equipping learners, irrespective of their socio-economic background, race, gender, physical ability or intellectual ability, with the knowledge, skills and values necessary for self-fulfilment, and meaningful participation in society as citizens of a free country;
    - providing access to higher education;
    - facilitating the transition of learners from education institutions to the workplace; and
    - providing employers with a sufficient profile of a learner's competences.
- The National Curriculum Statement Grades R - 12 is based on the following principles:
    - Social transformation: ensuring that the educational imbalances of the past are redressed, and that equal educational opportunities are provided for all sections of the population;
    - Active and critical learning: encouraging an active and critical approach to learning, rather than rote and uncritical learning of given truths;
    - High knowledge and high skills: the minimum standards of knowledge and skills to be achieved at each grade are specified and set high, achievable standards in all subjects;
    - Progression: content and context of each grade shows progression from simple to complex;
    - Human rights, inclusivity, environmental and social justice: infusing the principles and practices of social and environmental justice and human rights as defined in the Constitution of the Republic of South Africa. The National Curriculum Statement Grades R – 12 is sensitive to issues of diversity such as poverty, inequality, race, gender, language, age, disability and other factors;
    - Valuing indigenous knowledge systems: acknowledging the rich history and heritage of this country as important contributors to nurturing the values contained in the Constitution; and
    - Credibility, quality and efficiency: providing an education that is comparable in quality, breadth and depth to those of other countries.
- The National Curriculum Statement Grades R - 12 aims to produce learners that can:
    - identify and solve problems and make decisions using critical and creative thinking;
    - work effectively as individuals and with others as members of a team;
    - organise and manage themselves and their activities responsibly and effectively;
    - collect, analyse, organise and critically evaluate information;
    - communicate effectively using visual, symbolic and/or language skills in various modes;
    - use science and technology effectively and critically showing responsibility towards the environment and the health of others; and
    - demonstrate an understanding of the world as a set of related systems by recognising that problem solving contexts do not exist in isolation.
- Inclusivity should become a central part of the organisation, planning and teaching at each school. This can only happen if all teachers have a sound understanding of how to recognise and address barriers to learning, and how to plan for diversity.

The key to managing inclusivity is ensuring that barriers are identified and addressed by all the relevant support structures within the school community, including teachers, District-Based Support Teams, Institutional-Level Support Teams, parents and Special Schools as Resource Centres. To address barriers in the classroom, teachers should use various curriculum differentiation strategies such as those included in the Department of Basic Education's *Guidelines for Inclusive Teaching and Learning* (2010).

1.4 TIME ALLOCATION

1.4.1 Foundation Phase

(a) The instructional time in the Foundation Phase is as follows:

Subject	Grade R (Hours)	Grades 1-2 (Hours)	Grade 3 (Hours)
Home Language	10	7/8	7/8
First Additional Language		2/3	3/4
Mathematics	7	7	7
Life Skills	**5**	**5**	**5**
• Beginning Knowledge	(1)	(1)	(1,5)
• Creative Arts	(1,5)	(1,5)	(1,5)
• Physical Education	(1,5)	(1,5)	(1)
• Personal and Social Well-being	(1)	(1)	(1)
Coding and Robotics	(1)	(1)	(2)
Total	**23**	**23**	**25**

(b) Instructional time for Grades R, 1 and 2 is 23 hours and for Grade 3 is 25 hours.

(c) Ten hours are allocated for languages in Grades R-2 and 11 hours in Grade 3. A maximum of 8 hours and a minimum of 7 hours are allocated for Home Language and a minimum of 2 hours and a maximum of 3 hours for Additional Language in Grades R – 2. In Grade 3 a maximum of 8 hours and a minimum of 7 hours are allocated for Home Language and a minimum of 3 hours and a maximum of 4 hours for First Additional Language.

(d) In Life Skills Beginning Knowledge is allocated 1 hour in Grades R – 2 and 2 hours as indicated by the hours in brackets for Grade 3.

1.4.2 Intermediate Phase

The instructional time in the Intermediate Phase is as follows:

Subject	Hours
Home Language	6
First Additional Language	5
Mathematics	6
Natural Sciences	2,5
Social Sciences	3
Life Skills	**3**
• Creative Arts	(1)
• Physical Education	(1)
• Personal and Social Well-being	(1)
Coding and Robotics	2
Total	**27,5**

1.4.3 Senior Phase

(a) The instructional time in the Senior Phase is as follows:

Subject Choice: Option 1	Subject Choice: Option 2	Hours
Home Language	Home Language	5
First Additional Language	First Additional Language	4
Mathematics	Mathematics	4,5
Natural Science	Natural Science	3
Social Sciences	Social Sciences	3
*Technology	*Economic Management Sciences	2
Coding and Robotics	Coding and Robotics	2
Life Orientation	Life Orientation	2
Creative Arts	Creative Arts	2
Total		**27,5**

* Schools/Learners can follow Option 1 (MST Stream) or Option 2 (Business Stream)

1.4.4 Grades 10-12

(a) The instructional time in Grades 10-12 is as follows:

Subject	Time allocation per week (hours)
I. Home Language	**4.5**
II. First Additional Language	**4.5**
III. Mathematics	**4.5**
IV. Life Orientation	**2**
V. A minimum of any three subjects selected from Group B Annexure B, Tables B1-B8 of the policy document, *National policy pertaining to the programme and promotion requirements of the National Curriculum Statement Grades R – 12,* subject to the provisos stipulated in paragraph 28 of the said policy document.	**12 (3x4h)**

The allocated time per week may be utilised only for the minimum required NCS subjects as specified above and may not be used for any additional subjects added to the list of minimum subjects. Should a learner wish to offer additional subjects, additional time must be allocated for the offering of these subjects.

2 SECTION 2: DEFINITION, AIMS, SKILLS AND CONTENT

2.1 INTRODUCTION

Coding and Robotics represents an interdisciplinary and multidisciplinary subject that integrates various components of STEAM (Science (including Computer Science), Technology, Engineering, Arts, and Mathematics).

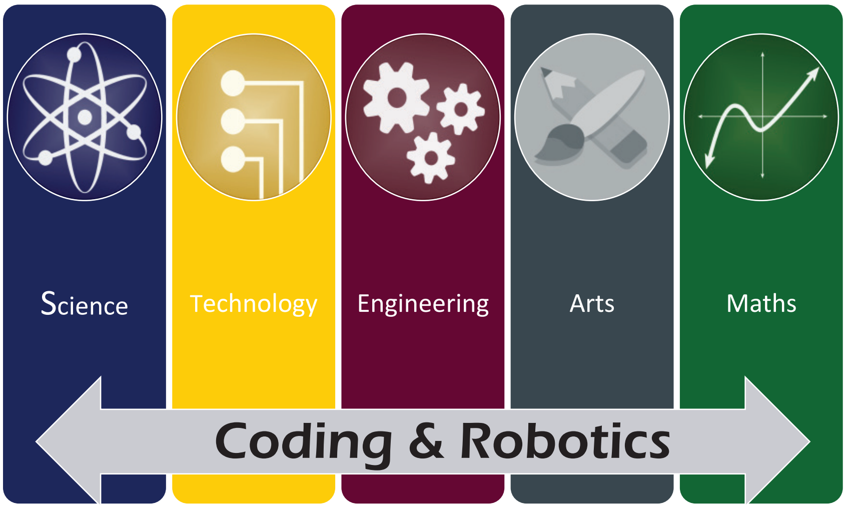


Figure 2.1 Coding and Robotics as a STEAM discipline

The main driving force behind the uptake and surge of Coding and Robotics as a subject at school level is the link to the 4th and 5th industrial revolution (4IR, and 5IR). In the context of this curriculum the focus resides in the grounding concepts of STEAM related subjects.

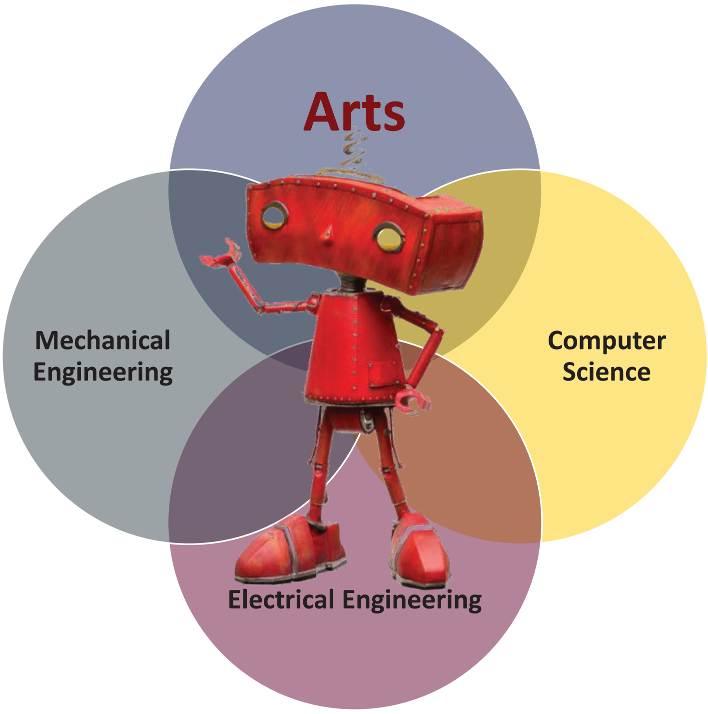


Figure 2.2: Coding and Robotics as a multi-disciplinary subject

2.2 What is Coding and Robotics

Coding and Robotics combine the principles of programming with the design, construction, and operation of robots. Programming concepts, practices, and perspectives are applied to control devices to perform specific tasks. It includes digital concepts that refer to various ideas, principles and processes that are associated with digital technologies and their use.

The Coding and Robotics curriculum is based on the following pillars as depicted in the figure below.

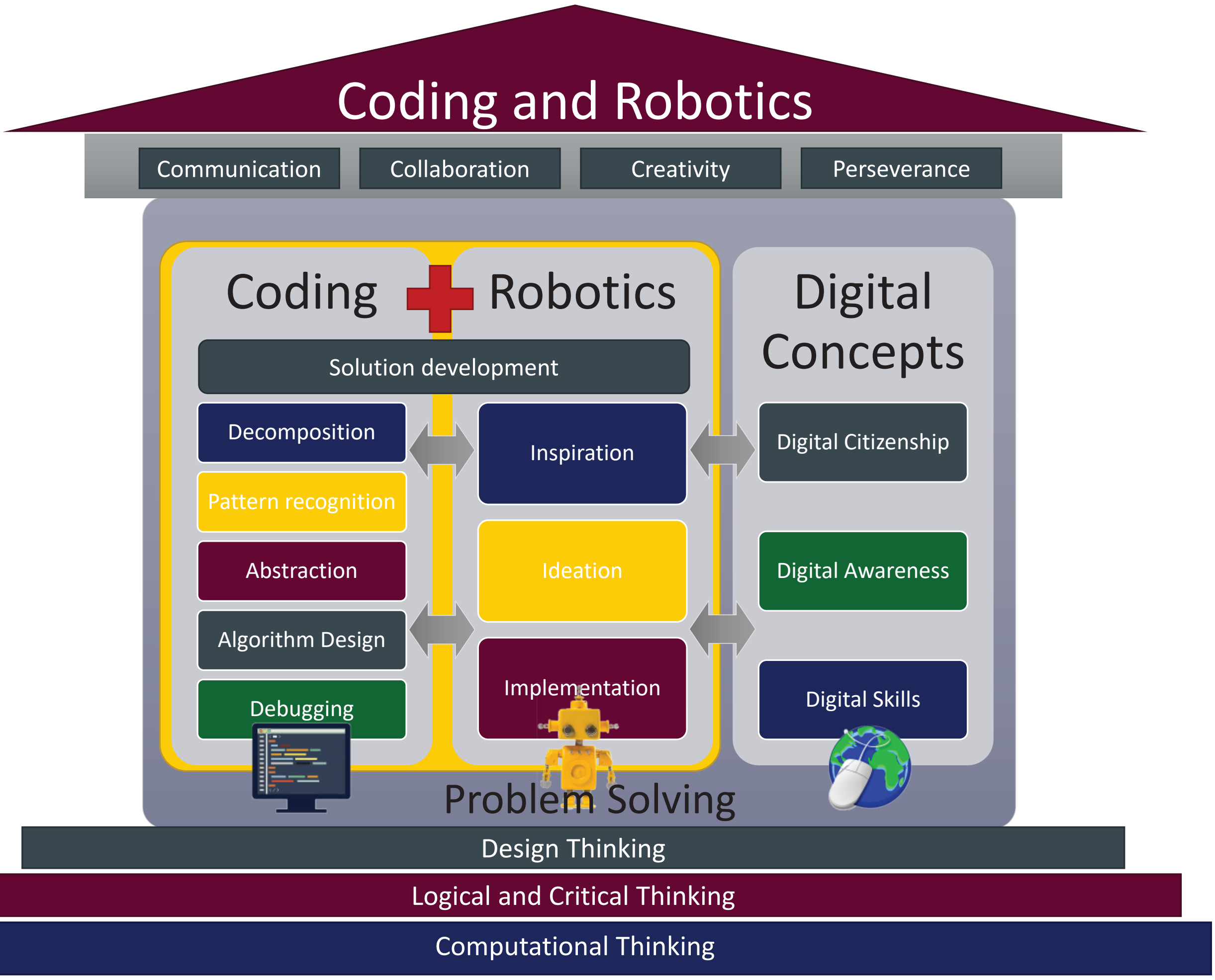


Figure 2.3: Overview of Coding and Robotics as a Subject

Coding is the process of creating a logical set of instructions that a human or a computing device can understand and execute, which require a deep understanding of computational thinking and problem solving.

Robotics deals with the design, operation, and use of devices and robots that can be programmed to perform tasks autonomously or semi-autonomously or by direct control. It presents the learners with the opportunity to see their thinking, design, and code in action.

Digital concepts encompass a range of digital literacy skills and awareness that enables learners to leverage digital technologies to their fullest potential and use digital tools responsibly.

2.3 Specific Aims

The teaching and learning of Coding and Robotics (C&R) aim to develop the following for the learner to be able to:

- develop computational thinking skills to solve problems.
- advance design thinking to develop creative and human-centred approaches to solve problems.

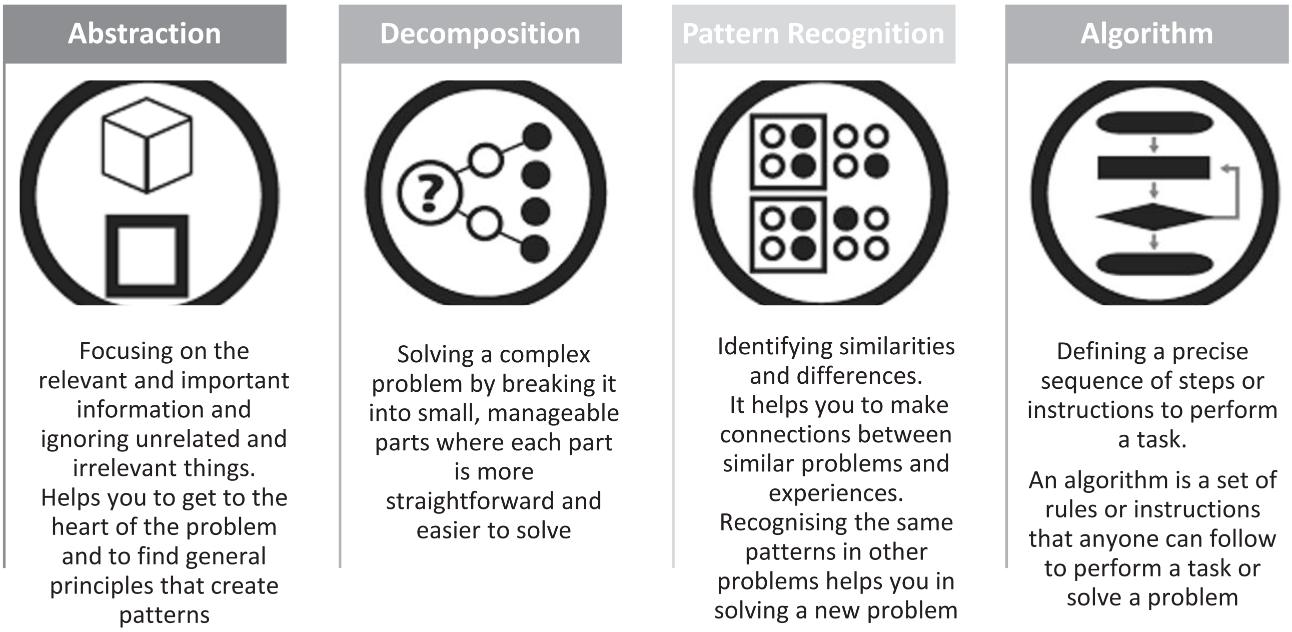
- become part of a generation of creative, innovative systems thinkers that can use coding, robotics, and digital competencies to express their ideas.
- foster creativity, critical thinking, collaboration, communication, and innovation.
- function ethically and effectively in a digital and information-driven world.
- develop a critical awareness of how technologies impact society at large.
- instil self-efficacy and confidence to deal with situations requiring computational thinking, design thinking and problem solving.
- prepare for future careers in STEAM related fields.
- adopt a culture of being self-directed, life-long learners who can apply their skills in a wide range of contexts and situations (adaptable, flexible and resilient).

2.4 SPECIFIC SKILLS

The following skills are specifically emphasised:

2.4.1 Computational Thinking

Computational thinking is an attitude and a skill set where one uses specific techniques and strategies to complete tasks successfully and to solve problems systematically. It further helps one in arriving at a solution that both humans and a computer can understand.



Focusing on the relevant and important information and ignoring unrelated and irrelevant things. Helps you to get to the heart of the problem and to find general principles that create patterns

Solving a complex problem by breaking it into small, manageable parts where each part is more straightforward and easier to solve

Identifying similarities and differences. It helps you to make connections between similar problems and experiences. Recognising the same patterns in other problems helps you in solving a new problem

Defining a precise sequence of steps or instructions to perform a task.

An algorithm is a set of rules or instructions that anyone can follow to perform a task or solve a problem

Figure 2.4: Computational Thinking Pillars

In Coding and Robotics, computational thinking helps learners to develop problem-solving strategies which they can apply when developing coding solutions (algorithms) as well as robotics solutions. It can also be applied to solve everyday life.

In terms of robotics, learners are demonstrating computational thinking concepts and practices when designing, constructing, and programming a robot. The robot's performance demonstrates the result of the learner's computational thinking practices as they iteratively test and debug their coding.

2.4.2 Design Thinking

In education, design thinking (DT) refers to a human-centred approach that encourages creativity and innovation when generating user-focused products, services, or experiences. Design Thinking is often expressed as an activity that involves the three **I** processes, namely:

- **Inspiration:** where creative thinking is applied to tackle a problem or challenge at hand, by gaining a deeper understanding of the problem and its context as well as to identify opportunities for innovation.
- **Ideation:** involves the generation of a wide range of ideas and potential solutions using various approaches such as brainstorming, prototyping and experimentation.
- **Implementation:** where the ideas and potential solutions are put into action. It includes testing, getting feedback and subsequent improvements of the design or solution.

Related to the three **Is**, is the notion that Design Thinking is also a problem-solving approach that combines creativity with structure and human-centred methods to understand and tackle challenges which involves empathizing with users, defining their needs, ideating possible solutions, prototyping, and testing those solutions, and iterating based on feedback. The following describes the design process:

- **Empathise:** involves gaining an understanding of who the end user is in a specific context, and how the envisaged solution will be appropriate towards addressing the problem.
- **Define:** relates to specifying in detail what the users' needs are, which could include the goals, skills available, and core principles that will guide the work to be done.
- **Ideate:** pertains to the creation of ideas and solutions using techniques such as brainstorming.
- **Prototype:** concerns the creation of one or several solutions to address the problem at hand.
- **Test:** relates to the process of determining how well the solution solves or address the problem. In this phase, feedback is important as the feedback could be used towards the improvement and enhancement and/or redesign of the complete solution or artefact.

Figure 2.6 depicts the relationship between the Design Thinking and Design Problem Solving approach.

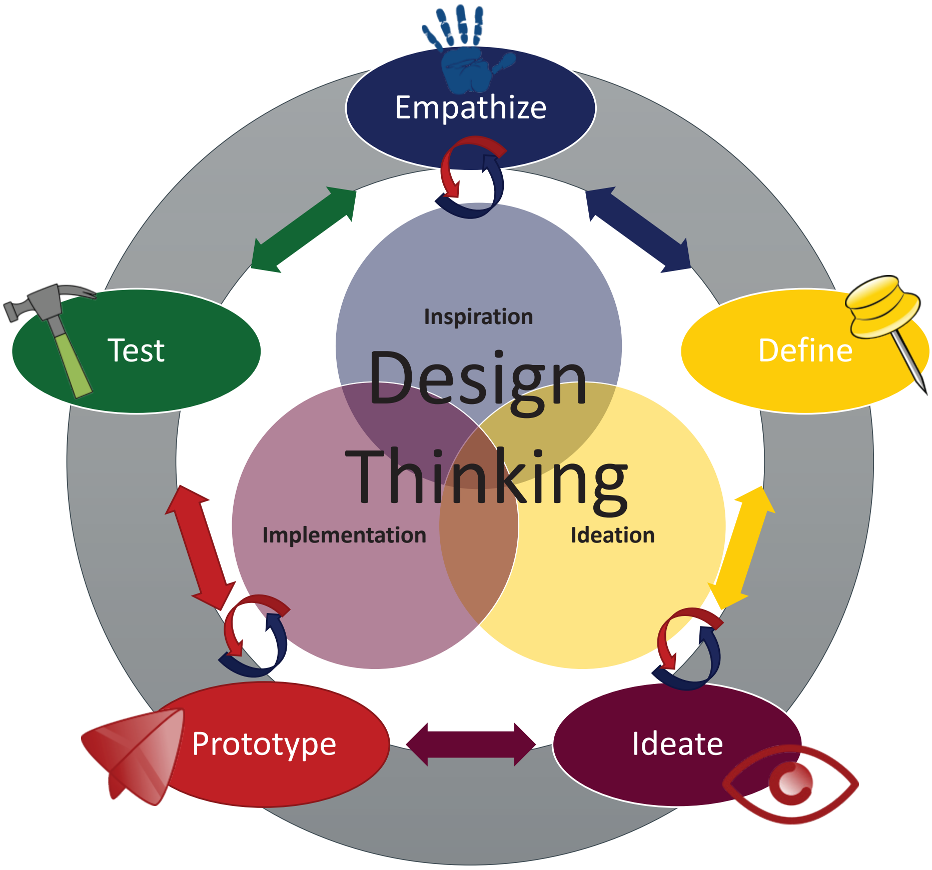
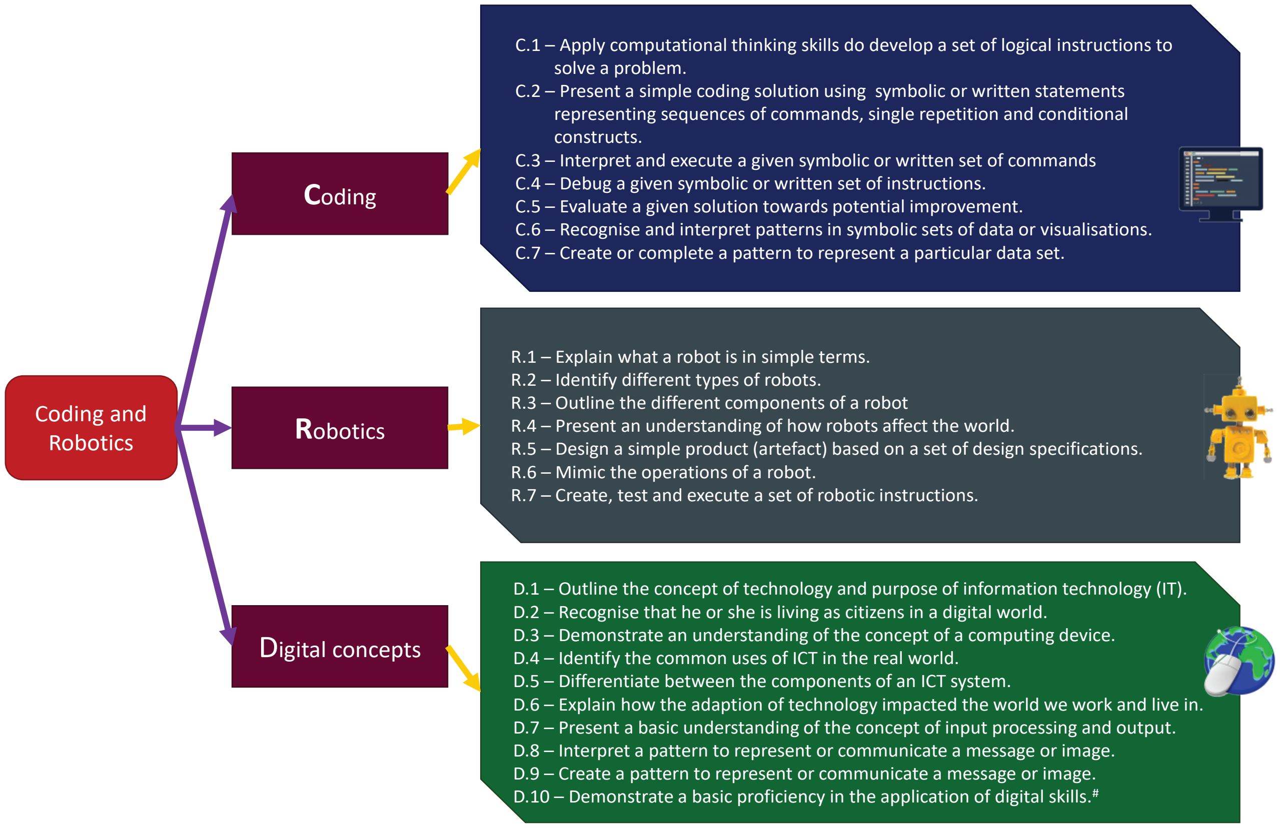


Figure 2 5: Design Thinking and Problem-Solving Process

2.5 HIGH-LEVEL COMPETENCIES – CODING AND ROBOTICS

The three main topical areas of Coding and Robotics each comprises a set of key learning competencies central to their area of focus.

The following diagram outlines the three main topical areas and the main learning competencies, associated at the final stage of curriculum cognition wherein the learner demonstrates competence and proficiency at the appropriate level.



Coding and Robotics

Coding

C.1 – Apply computational thinking skills do develop a set of logical instructions to solve a problem.
C.2 – Present a simple coding solution using symbolic or written statements representing sequences of commands, single repetition and conditional constructs.
C.3 – Interpret and execute a given symbolic or written set of commands
C.4 – Debug a given symbolic or written set of instructions.
C.5 – Evaluate a given solution towards potential improvement.
C.6 – Recognise and interpret patterns in symbolic sets of data or visualisations.
C.7 – Create or complete a pattern to represent a particular data set.

Robotics

R.1 – Explain what a robot is in simple terms.
R.2 – Identify different types of robots.
R.3 – Outline the different components of a robot
R.4 – Present an understanding of how robots affect the world.
R.5 – Design a simple product (artefact) based on a set of design specifications.
R.6 – Mimic the operations of a robot.
R.7 – Create, test and execute a set of robotic instructions.

Digital concepts

D.1 – Outline the concept of technology and purpose of information technology (IT).
D.2 – Recognise that he or she is living as citizens in a digital world.
D.3 – Demonstrate an understanding of the concept of a computing device.
D.4 – Identify the common uses of ICT in the real world.
D.5 – Differentiate between the components of an ICT system.
D.6 – Explain how the adaption of technology impacted the world we work and live in.
D.7 – Present a basic understanding of the concept of input processing and output.
D.8 – Interpret a pattern to represent or communicate a message or image.
D.9 – Create a pattern to represent or communicate a message or image.
D.10 – Demonstrate a basic proficiency in the application of digital skills.#

Figure 2 6: High-level Curriculum Competencies

A competence is a combination of knowledge skills, attitudes, and values which is reflected in behaviour that can be observed, measured, and evaluated. It refers to the ability to perform a specific task successfully and efficiently or in a manner that yields desirable outcomes.

2.6 CODING AND ROBOTICS CONCEPTS, PRACTICES AND PERSPECTIVES

2.6.1 Coding

In coding, the following concepts, practices, and perspectives must be developed and practised repeatedly:

Concepts

- Algorithm
- Sequence
- Loop (Iteration)
- Conditional (Decisions)
- Operator
- Logic
- Data
- Event
- Debug
- Representation
- Parallelism
- Automation

Practices

- Abstraction
- Decomposition
- Pattern Recognision
- Generalisation
- Algorithm Design
- Incremental Development
- Testing and Debugging
- Evaluation
- Modularise
- Logical thinking
- Creating computational artefacts

Perspectives

- Expressing and Creating
- Questioning
- Connecting
- Collaboration
- Perserverance
- Choice of Conduct

Figure 2.7 Coding Concepts, Practices and Perspectives

The following table describes the coding content to be covered and the skills to be developed. Table 2-1 must be read in conjunction with Table 2-4 in Section 2.12.1 and Table 2-7 in Section 2.16.1 for competencies and progression per grade.

Table 2-1 Coding content and skills

Concept	Content/Skills
Algorithm	• Definition and importance of algorithms in computer programming. • Characteristics of a good algorithm. • Examples of algorithms in everyday tasks and programming. • Algorithm development using computational thinking.
Sequence	• Understanding the concept of sequential execution of instructions. • Introduction to basic programming constructs like statements and expressions. • Writing simple programs to perform sequential tasks.
Loop (iteration)	• Explanation of loops and their purpose in programming. • Different types of loops (e.g., while loop, for loop) and their syntax. • Examples demonstrating the use of loops for repetitive tasks. • Writing simple programs to perform tasks that include repetition
Conditional (Decision)	• Understanding conditional statements (if…then, if…then…else)) and their role in decision-making. • Using comparison operators in conditional statements. • Writing programs with conditional logic to handle different scenarios.
Operator	• Assignment, comparison, and logical operators. • Precedence and associativity rules for operators. • Use of operators in expressions and assignments in programs
Logic	• Introduction to Boolean logic and truth tables. • Understanding logical operators (AND, OR, NOT) and their application in programming. • Writing programs that implement logical operations and evaluate conditions.
Data	• Types of data (e.g. numbers, string, Boolean) and their use in programming. • Variables and data types. • Input/output and processing operations for data manipulation.
Event	• Concept of events and event-driven programming. • Handling user interactions and system events in programs. • Implementing event handlers in programs.
Debug	• Techniques for identifying and fixing errors in code (debugging). • Using debugging tools and techniques (e.g., trace tables). • Debug common programming errors (syntax errors, logic errors).
Representation	• Understanding data representations (binary). • Exploring the concept of abstraction in programming.
Automation	• Exploring automation concepts and their significance. • Writing scripts to automate repetitive tasks.
Parallelism	• Concept of parallelism using e.g. two scripts, running concurrently, allowing different actions to happen simultaneously (e.g. broadcast & receiving, clones, parallel blocks, e.g. "forever" block can run continuously while other code executes concurrently.

2.6.2 Robotics

In addition to the coding concepts, practices and perspectives, in robotics, the following concepts, practices, and perspectives must be developed and practised repeatedly:

Concepts

- Motion
- Sensor
- Actuator
- Controller
- Logic
- Power Source
- Automation
- Instruction
- Communication
- Coding (Programming)

Practices

- Computational Thinking
- Design Thinking
- Prototyping
- Design and Construction
- Algorithm Design
- Testing and Reconfiguration
- Reflection and Iteration
- Creative Thinking
- Logical thinking
- Creating robotics artefacts

Perspectives

- Expressing and Creating
- Innovation
- Questioning
- Connecting
- Collaboration
- Perserverance
- Choice of Conduct

Figure 2.8 Robotics Concepts, Practices and Perspectives

The following table describes the robotics content to be covered and the skills to be developed. Table 2-2 must be read in conjunction with Table 2-5 in Section 2.12.2 and Table 2-8 in Section 2.16.2 for competencies and progression per grade

Table 2-2 Robotics content and skills

Concept	Content/Skills
Motion	• Introduction to different types of robot motion: linear, rotational, and combined. • Exploring methods of locomotion such as wheels, tracks, legs, and aerial mechanisms.
Sensor	• Overview of sensors used in robotics, including proximity sensors, cameras, ultrasonic, microphone, temperature sensors. • Explanation of sensor principles and how they gather data from the environment. • Applications of sensors in navigation, obstacle avoidance, object detection, and environmental monitoring.
Actuator	• Introduction to actuators responsible for converting electrical energy into mechanical motion. • Types of actuators: electric motors (DC motors, servo motors). • Understanding the role of actuators in robot manipulation, locomotion, and control.
Controller	• Components as part of a robot responsible for controlling the robot, gathering input, and providing output. (Examples: Arduino, Raspberry Pi, Micro: bit)
Logic	• Introduction to logical operations and decision-making in robotics. • Understanding Boolean logic and its application in robot control. • Implementing logical operations for conditional behaviour, state transitions, and autonomous decision-making.
Power Source	• Overview of power sources for robotics, including batteries, and external power supplies (e.g. solar). • Understanding power requirements and considerations for selecting appropriate power sources. • Designing power distribution systems and managing power consumption for optimal robot performance.
Automation	• Explanation of automation in robotics as the process of performing tasks with minimal human intervention. • Applications of automation in manufacturing, logistics, agriculture, healthcare, and service industries. • Designing automated systems using robots for repetitive, dangerous, or labour-intensive tasks.
Instruction	• Understanding instructions as commands given to robots to perform specific actions. • Types of instructions: sequential instructions, conditional instructions, repetitive instructions (loops). • Writing clear and precise instructions for programming robots to accomplish desired tasks.
Communication	• Basic overview of communication technologies used between two or more devices, e.g. Wi-Fi, Bluetooth.
Coding	• Introduction to a robotics programming environment. • Basics of robot programming: variables, data types, control structures (sequence, if statements, loops), functions, and libraries. • Hands-on coding exercises and projects to develop skills in algorithm development and in robot programming.

2.6.3 Digital Concepts

Digital concepts are fundamental ideas and principles that underpin and support coding and robotics. They encompass various aspects of technology and computer science, providing the context and application for these fields. In Coding and Robotics, digital concepts are divided into the following topics: Digital Citizenship, Digital Awareness and Digital Skills. The following must be read in conjunction with Table 2-6 in Section 2.12.3.

2.6.3.1 Digital Citizenship

The rights, responsibilities, and behaviours (respect, integrity, and safety) displayed by individuals in the digital world. It encompasses a spectrum of behaviours, spanning from respecting the privacy of others to protecting personal data, being mindful of online threats, and ensuring one's safety in the digital sphere.

Responsible behaviour	Information Assessment	Impact and Responsibility
•Implications of digital citizenship •Responsible and ethical behaviour in the digital realm. •Cybersecurity awareness such as protecting personal information online, recognising cyber threads and practiaing safe online behaviour. •Privacy & security (strong passwords, not sharing personal information), cyberbullying, digital footprint and netiquette. •Digital health and welfare.	•Credibility and reliability of online sources and information. •Fake news •Intellectual property and its implications.	•Awareness of how technology adaptation influence our work and lifestyle •Consequences and implications of online actions. •Lasting impact of online content (digital footprint) •Ethical use of computers, including software and robotics applications. •Dangers of the online world, computer/cyber crime

Figure 2-9 Digital Citizenship Concepts

Digital Citizenship helps to develop an awareness of responsible and ethical behaviour in the digital world as it provides principles for shaping the digital landscape and influencing individual and collective behaviour online. In Coding and Robotics, understanding digital citizenship is important when developing software and robotics applications to ensure they are used in a responsible and ethical manner.

2.6.3.2 Digital Awareness

The recognition of the competencies, expertise, and the mindset needed by individuals effectively to use digital tools, entail understanding and the applications of technologies in a world that is becoming more interconnected. This emphasises the essential sense of familiarity, adaptability, and proficiency needed for utilising fundamental technology.

Data and Information	Computing devices	Networks and Communication
•Data collection, storage (including cloud storage) and processing •Transformation of data into information. •Utilisation of data and information for decision-making and innovation. •POPIA	•Technology vs Information Technology (IT) •Basic model of a computing device (input, processing, output and storage). •Different types of computing devices in Coding and Robotics. •Hardware components for input, processing and output. •Computing devices and their purpose (including basic network devices). •Hardware vs Software •Interaction between harware and software.	•Networks, basic components, role of network devices, media (wired and wireless), data transmission. •Internet and the role of the Internet of things (IoT). •Information and Communication Technology (ICT) - components and real-world uses. •Internet vs World Wide Web (WWW). •Websites, webpages, URL, hyperlink, browser, search engine

Figure 2-10 10 Digital Awareness Concepts

2.6.3.3 Digital Skills

An essential set of a range of abilities that enable individuals to effectively use digital devices, software, and platforms to perform various tasks.

Application Skills	Patterns and Communication	Digital Literacy
•Use and manage applications used in Coding and Robotics (software environment). •File and Folder management •Input, processing and output in computing and robotics •Use an application such as Paint to create backgrounds and sprites •Use presentation software to communicate information	•Patterns in coding and robotics for communication, including data analysis, visualisation and conveying messages or information. •Boolean logic (including truth tables and logic gates).	•Find, evaluate and use informaiion effectively and ethically

Figure 2-11 Digital Skills Concepts

2.7 APPROACH TO TEACHING CODING AND ROBOTICS

Coding and Robotics, as a subject, is process-driven as it focuses on Coding and Robotics processes, rather than just exit skills or products. Coding develops cognitive and critical thinking skills as it emphasises the development of knowledge, skills, strategies, and attitudes that enable learners to become more effective individuals. Coding and Robotics also supports learners to develop metacognitive skills, which include planning, developing, testing, evaluation and reflecting.

2.7.1 Problem-based Learning

Teaching and learning will follow a problem-based learning approach. Problem-based learning (PBL) is an active and learner-centred approach to learning involving certain cognitive processes that aims to develop critical thinking, problem-solving, and collaboration skills. The goal of PBL is to help learners learn how to apply knowledge and skills using problems, rather than just memorising information for tests. PBL also encourages learners to ask questions and seek answers, rather than passively receiving information. It also supports the development of self-directed learning.

In Senior Phase, learners will be given small, manageable problems which they need to solve using a problem-solving process. To develop and enhance self-efficacy (the learner's belief that he/she will be able to complete the task or solve the problem), the challenge of the task or problem should match the learners' competencies.

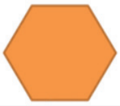
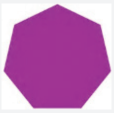
Example of a manageable problem and algorithm development using computational thinking and the problem-solving process in Senior Phase:

Problem:
Draw a geometric figure for the number of sides provided by the user.

Step 1: Understand the problem.

When the user enters a number (3 – 360), the program must draw a geometric figure with the number of sides entered, e.g. if the user enters 4, the program must draw a square, if the user enters 6, the program must draw a hexagon, etc.

Step 2: Analyse the problem

Shape	Square	Pentagon	Hexagon	Heptagon	Octagon
					
Number of sides	4	5	6	7	8
Number of angles	4	5	6	7	8

Each figure has the same number of angles than the number of sides

Step 3: Break down the problem into smaller, more manageable problems / use experience from similar problems.

Decompose (break down) the problem, into more familiar ones, using **abstraction.**

From previously drawing a square, we know

Repeat 4 times
    Turn right (90 degrees)
    Move 100 steps

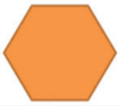
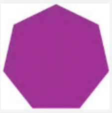
From previously drawing a triangle

Repeat 3 times
    Turn right 60 degrees
    Move 100 steps

Abstraction and decomposition
Break down the problem into smaller, more manageable parts and simplify ideas by focusing on essential details. to understand the underlying structure.

Iterations for drawing each shape are the same as the number of sides (or angles)

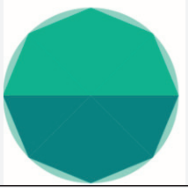
Step 4: Look for patterns (Pattern recognition)

Shape	Square	Pentagon	Hexagon	Heptagon	Octagon
					
Number of sides	4	5	6	7	8
Number of angles	4	5	6	7	8
Size of interior angle	90°	72°	60°	51,4°	45°
Sum of interior angles	360°	360°	360°	360°	360°

Pattern recognition involves finding the similarities or patterns among small, decomposed problems that can help us solve more complex problems more easily and efficiently.

  - As the number of sides **increases**
  - the size of the interior angles **decreases**
  - the form of the shape gets closer to a circle
- The **sum** of the interior angles of a shape equals **360°** and
- A **circle** is also 360°
- The size of each the interior angle is 360 divided by the number of sides/angles

Octagon within a circle



Step 5: Develop a high-level solution or algorithm (abstraction)

Draw a geometric shape:

Step 1: Get the number of sides.

Step 2: Determine the size of the interior angle.

Step 3: Determine the length of the sides (number of moves/steps)

Step 4: Draw the shape

We can now break the problem into ***four*** rather easy **sub-problems** or **main steps/ ideas (decomposition,** using **abstraction)**.

Abstraction also helps us to realise that we can ignore 3 of the rectangles / only focusing on 3 rectangles – one of each size to solve the

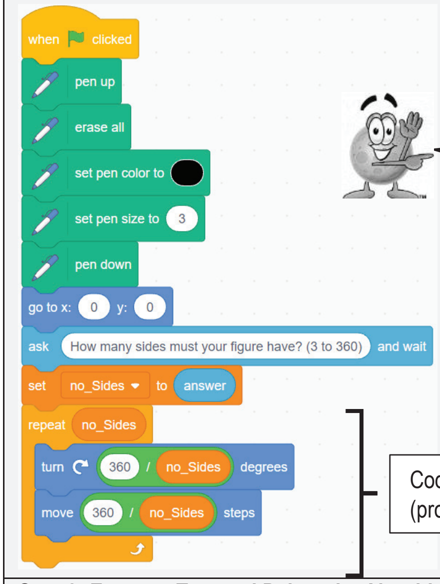
Step 6: Detailed Algorithm (Decomposition)

1. Step 1: Get the number of sides.
   - 1.1. Ask the user to enter the number of sides of the figure to be drawn
2. Step 2: Determine the size of the interior angle
   - 2.1. Divide 360 by the number of sides/angles.
3. Step 3: Determine the length of the sides
   - 3.1. Divide 360 by the number of sides/angle).
4. Step 4: Draw the shape
   - 4.1 Pen instructions for drawing
   - 4.2 Repeat number_of_sides
     - Turn 360/number_of_sides
     - Move 360/number_of_sides

Each step in the **high-level algorithm** was broken down into more specific, detailed steps, giving more detailed instructions.

Effective problem decomposition is a key skill in programming.

Step 7: Write the code



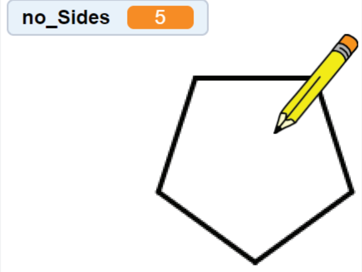
```
when 🏁 clicked
pen up
erase all
set pen color to ●
set pen size to 3
pen down
go to x: 0 y: 0
ask How many sides must your figure have? (3 to 360) and wait
set no_Sides to answer
repeat no_Sides
    turn ⟳ 360 / no_Sides degrees
    move 360 / no_Sides steps
```

Convert the algorithm to code, execute and test, debug if required

Input from user

Code for drawing the geometric figure (processing)

Output from computer



```
no_Sides 5
```

Step 8: Execute, Test and Debug the Algorithm.

- Did the program work correctly for different number of sides?
- If the answer is yes, the problem is solved, else you need to identify the error and fix the algorithm (debug).

Generally, problem-based learning

- enables learners to develop problem solving strategies as well as subject knowledge and skills.
- enables learners to be more engaged in learning.
- stimulates critical thinking.
- promotes self-directed learning as learners generate problem-solving strategies.
- promotes metacognition as learners compare and reflect on solutions.
- assesses learning in ways which demonstrate understanding and competency.

See **Section 4.2** for problem-based learning assessment guidance.

PBL could incorporate strategies such as cooperative learning where learners work in small groups to solve a coding or robotics problem or use pair programming where learners work in pairs to solve coding or robotics problems.

2.7.2 Cooperative Learning

Cooperative learning is an active teaching-learning strategy where learners work in small groups, they help each other learn and in doing so, increase their joy and skills in the learning process.

Learning activities and roles are structured and overseen by the teacher, and each member of the group oversees the academic performance of the others. To successfully implement cooperative learning, leading authors in the field (David Johnson and Roger Johnson) emphasise the intentional stimulation of five basic elements (Johnson & Johnson, 2021:55-56) namely:

- **Positive interdependence**: Learners should feel like they are linked in such a way that one cannot succeed unless all in the group succeeds. Teachers should thus find ways of stimulating positive interdependence in their group activities – one possibility is giving learners different roles to fulfil; hence the group cannot move forward unless all roles are successfully fulfilled.
- **Individual accountability**: Learners should know that all will be assessed individually as well. "*The purpose of cooperative learning groups is to make each member a stronger individual in his or her right*". One way of stimulating individual accountability is by giving learners individual marks for how well they contributed to the group activity – this assessment can occur either via teacher assessment or peer assessment – by doing this, everyone will know that they cannot get a freeride during the group activity as their inputs are also individually assessed.
- **Promotive interaction**: Learners' successes are increased due to the sharing of resources, support provided, and praise and encouragement given by their group members. Teachers thus need to stimulate promotive interaction which can be done by giving different resources to different learners. Giving learners different roles also stimulate promotive interaction.
- **Social skills**: Stimulating social skills becomes an intentional endeavour of the teacher. Teachers could provide learners with resources on how to effectively form part of a team, how to communicate well and how to resolve conflict, should it arise.
- **Group processing**: Group processing forms part of reflection during and after the group activity. Teachers can stimulate group processing by giving learners a reflection sheet or by asking them open-ended questions to stimulate reflective conversations. Questions such as: "What worked well during your group activity"? or "*Describe the best experiences and worst experiences of the group activity*".

Cooperative learning can improve the learner's performance and teaches the value of teamwork, cooperation, communication, self-denial, and initiative taking.

2.7.2.1 Implementing cooperative learning in Senior Phase Coding and Robotics

Example of cooperative learning activity for Senior phase learners on the topic of robotics (Grade 7 (C.3)): "*A simple set of commands in relation to R.6, are correctly executed physically, on paper or with an educational tool.*"

Task: Determine where a robot (simulated by one of the learners) will end after executing a set of instructions, including at most nine steps, provided in an algorithm.

Divide the class into groups of four. Two learners could take on the roles of instructor and interpreter respectively and the other two learners the roles of **robot** and **debugger**.

- **Instructor:** Reads out the steps from the algorithm
- **Interpreter**: Puts steps from algorithm into "layman's English" /Explain steps in plain English
- **Robot**: Executes the steps from the interpreter
- **Debugger**: Evaluates the movement of the robot to determine whether it executed steps correctly.

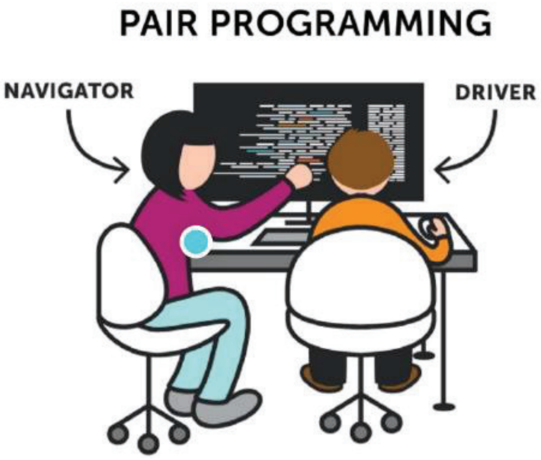
Tools that can be used to develop the algorithm: pen-and-paper, coding cards (e.g. Tanks) for algorithm and interpretation, then, code algorithm using blocks from block-based coding platform, implement, test and debug in block-based coding environment.

Refer to Annexure B for cooperative learning assessment guidance.

Pair programming could also be used as a cooperative teaching and learning strategy to solve programming problems.

2.7.3 Pair Programming

Pair programming is a pedagogical approach that involves two learners working together on one computer or one piece of paper to complete a shared goal/task. It emanates from the programming industry yet has proven to be successful even at school level. One of the learners fulfils the role of the "*driver*" while the other learner fulfils the role of the "*navigator*".



The driver is the learner who may use the computer and handles the keyboard, or draws on the paper and handles the pencil, whereas the navigator is the learner who utilises the resources, and reviews the driver's work throughout, providing feedback and suggestions to the driver, pointing out errors and asking questions of the teacher. Pair programming is a collaborative effort that involves a lot of communication, discussion, and problem-solving.

Although pair programming can be implemented as a collaborative "unstructured" pair activity, it is best to stimulate the five basic elements of cooperative learning as described above, when implementing pair programming in the classroom.

It also appears particularly promising in situations where there are not enough computing devices for learners to work individually, as well as for increasing learning and engagement with technology by learners with limited device experience. It is also suggested that learners show higher confidence when programming in pairs. It allows learners to share knowledge and learn from each other, thereby improving the quality of the learning engagement.

Implementing pair programming in Senior Phase Coding and Robotics

Example of pair programming activity for Senior phase on the topic of Coding (C.1 and C.2):

Apply computational thinking skills to develop a set of logical instructions to solve a problem.

Learners are divided into pairs. Learners should draw a square using a set of instructions. One learner fulfils the role of "driver" and the other "navigator".

- **Driver** – The learner acting as the driver will be the one completing the steps in a block-based program and/or unplugged on a piece of paper.
- **Navigator** – The learner acting as the navigator may consult the textbook and/or other resources. They may also ask the teacher for help.

Learners need to find a way to draw a rectangle using a set of instructions. This implies having the drawing tool "turn" several degrees and moving forward a certain number of pixels/steps. Learners should be able to first work this out by "directing" each other and then put these instructions over into algorithm.

Note:
The teacher may swop the learners' roles as the activity progresses to ensure that both learners have a chance to fulfil each role. You may also ask any one of the learners to present their work to the class. This ensures that both learners feel a need to engage and gives more learners an opportunity to practice communication skills.

2.7.4 Deliberate Practise

A subject such as Coding and Robotics not only requires thinking skills, but also requires focused teaching and ample practise. This practise should, however, be purposeful, well thought through with gradual increase in complexity.

The curriculum is designed to encourage deliberate practise, as competencies are repeated within and across grades. The concept of deliberate practise is particularly focused on skill acquisition and development and is key in the development of competency and expertise in subjects such as coding.

Deliberate practise is a specific type of practise that involves setting specific goals, receiving feedback, and making focused efforts to acquire and improve skills and performance. It is not simply repeating skills over-and-over again, but rather adjusting to improve competencies as well as gradually adding additional competencies that lead to mastery. It therefore involves purposeful repetition, feedback-driven metacognition, and extension to improve performance (Ericsson, 2008; Deans for Impact, 2016; Ericsson *et. al*., 2018).

In terms of extension, deliberate practice involves extending the amount of time spent practising, adding new features, and increasing the complexity of tasks. The goal is to push beyond one's comfort zone to achieve growth and improvement.

2.7.5 Science of Learning

Science of Learning, a multidisciplinary field combines research from cognitive psychology, neuroscience, educational psychology, and other related disciplines to understand how people learn. It also aims to identify the most effective teaching and learning strategies based on empirical evidence that has been shown to improve long-term retention of information and enhance learning outcomes.

Learning is an iterative process that requires that one continually revisits what one has learned earlier, update it, and connect it with new knowledge. Learning always builds on a store of prior knowledge and is the residue of thought. New learning requires a considerable amount of practise and meaningful connections to existing knowledge. Learning, therefore, requires learners thinking (Brown *et al*., 2014; Dereck Bok Center, Harvard University, 2023).

Science of learning includes the following learning strategies (*Weinstein et* al., 2018):

- **Retrieval practice**: Bringing learned information to mind from long-term memory.
- **Spaced practice**: Spreading learning activities out over time/reviewing previously learned information at gradually increasing intervals.
- **Interleaving**: Switching between topics while learning.
- **Examples**: When learning abstract concepts, illustrating them with various examples or experiences.
- **Dual coding**: For example, combining words with visuals.
- **Elaboration**: Classroom discussions that require learners to relate new material to what they already know and to recall previously learned information, including asking *why* and *how* questions with learners explaining in their own words.
- **Interactive activities**: Engage actively with learning material using activities that require one to retrieve (recall) previously learned information.

2.8 LINKING CODING AND ROBOTICS WITH OTHER SUBJECTS

Coding and Robotics concepts are linked to Language, Mathematics, Natural Sciences, Technology and Life Skills in the Senior Phase. These cross-cutting concepts should therefore be integrated into Coding and Robotics to enhance the learning experience.

For example, coding often involves mathematical concepts, such as logic, arithmetic, and geometry whilst Robotics combines coding with principles of physics, engineering, and materials science, highlighting the interdisciplinary nature of digital concepts and skills. Other examples are:

Algorithms involve sequencing and summarising in literacy and breaking down complex problems into simpler steps in mathematics.

Modularity: Involves breaking down tasks into manageable units in computer science, while in mathematics, it involves breaking down a complex problem into smaller, manageable parts.

Control structures in coding determine how a set of instructions are executed within a program, while heuristic thinking in mathematics involves using logical thinking and trial and error to solve problems.

Coding and natural language: The process of learning to code is also often likened to language acquisition, as learners progress through six distinct stages of understanding. These stages bear close resemblance to the stages of literacy development.

Design: Designing robotics artefacts links to aspects of Creative Arts as part of Life Skills.

Digital concepts: Aspects such as the impact of technology and being a digital citizen, links to Life Orientation.

By developing these skills in Coding and Robotics, learners can develop habits of mind that will be valuable in all other subjects.

2.9 TIME ALLOCATION

In Senior Phase, 2 hour per week (20 hours per term) is allocated for Coding and Robotics.

The following table provides the time allocation as a percentage of the total available time per term:

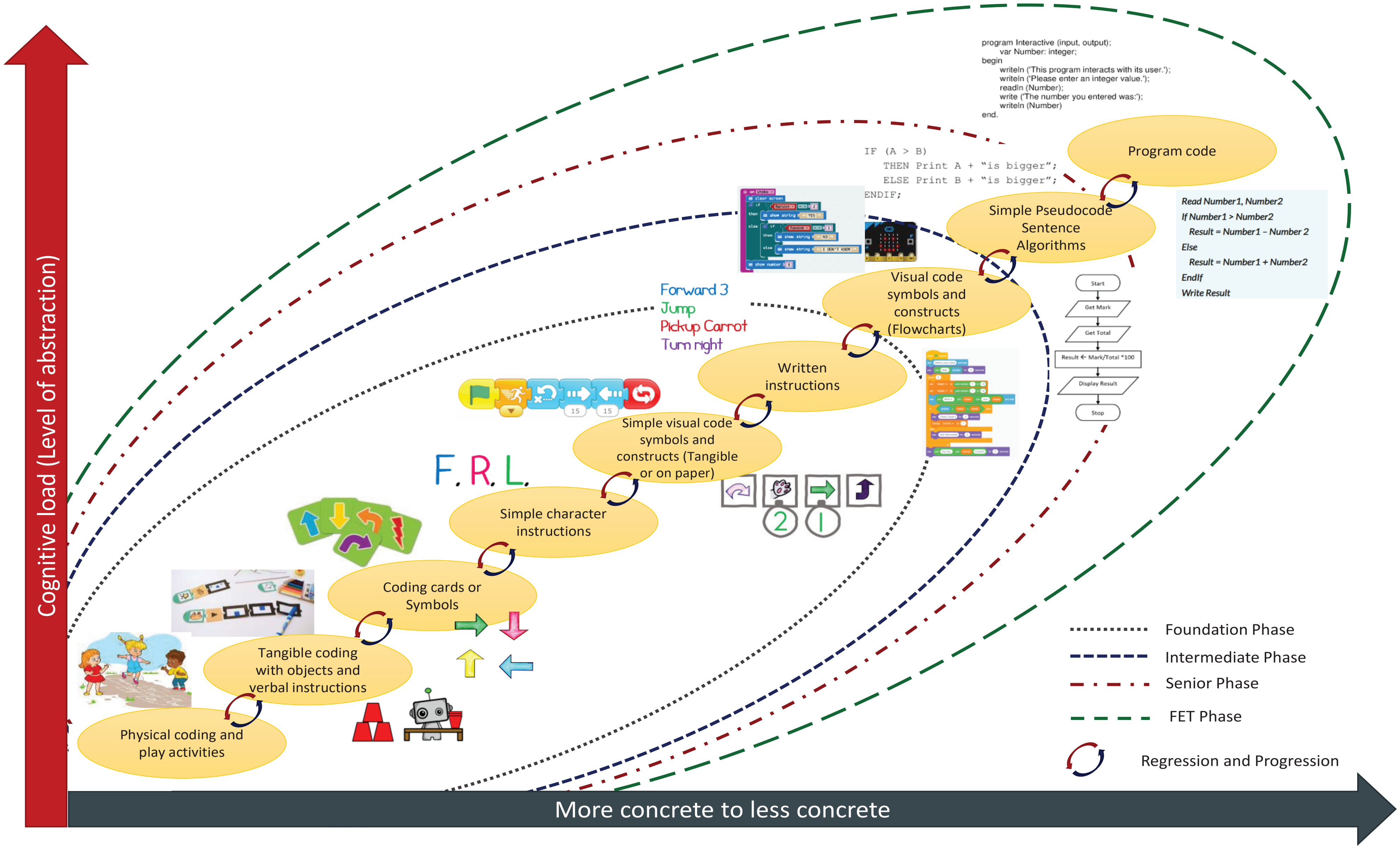
Table 2-3: Time allocation for Senior Phase Coding and Robotics

	Grade 4	Grade 5	Grade 6	Grade 7	Grade 8	Grade 9
Coding	50	50	50	50	45	45
Robotics	25	25	30	30	35	35
Digital Concepts	25	25	20	20	20	20

Available time should be allocated as indicated by the percentages in the table above.

Note:
Sections 2.12.1 (coding content) 2.12.2 (robotics content) and 2.12.3 (digital concepts content) are linked and support each other. Various competencies across the three strands can therefore be linked and dealt with in an integrated fashion. Section 3 (unpacking of the content) provides examples and notes and suggests pedagogical approaches.

2.10 RESOURCES REQUIRED TO OFFER CODING AND ROBOTICS IN SENIOR PHASE



```
program Interactive (input, output);
    var Number: integer;
begin
    writeln ('This program interacts with its user.');
    writeln ('Please enter an integer value.');
    readln (Number);
    write ('The number you entered was:');
    writeln (Number)
end.
```

```
IF (A > B)
  THEN Print A + "is bigger";
  ELSE Print B + "is bigger";
ENDIF;
```

```
Read Number1, Number2
If Number1 > Number2
  Result = Number1 – Number 2
Else
  Result = Number1 + Number2
EndIf
Write Result
```

Figure 2-12: Programming resources for Coding and Robotics

2.10.1 Coding Resources

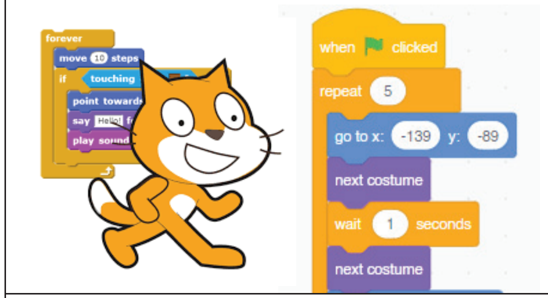
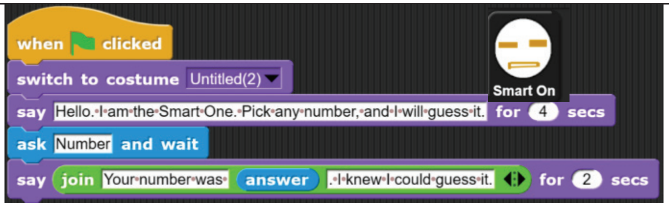
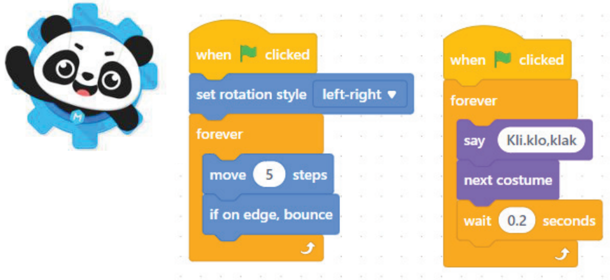
Refer to Figure 2.12:

In the Senior phase, learners will follow a block-based coding approach. It allows users, especially novices and kids, to create programs using visual elements like blocks and symbols instead of text. In this type of environment, users can drag and drop various blocks representing commands and snap them together like puzzle pieces to create their programs. It helps learners to the learn the foundational concepts and principals of coding without getting overwhelmed by the intricacies of text-based syntax and it minimises errors (such a formatting, punctuation, or spelling mistakes, semantics that could discourage learners) associated with the complex syntax of text-based environments. By abstracting away the textual complexities, it therefore reduces cognitive load and allows learners to focus on the problem and the foundational coding concepts as well as the underlying logic of their programs, rather than spending too much mental effort on getting the code syntax right. It therefore serves as an effective steppingstone for beginners to develop their problem-solving and programming skills before transitioning to more advanced coding languages and environments.

> "Computer programming is a highly cognitive skill, which requires mastery of multiple domains, and is acknowledged as being difficult to learn, making it essential to take into account the cognitive loads (CLs) imposed on learners, as well as their abilities to absorb this knowledge during the teaching and learning process". *Berssanette & de Francisco (2022).*

Research also suggests that teaching computational thinking with block-based coding, learners (1) achieved substantial learning gains in algorithmic thinking skills, (2) were able to transfer their learning from Scratch to a text-based programming context, and (3) achieved significant growth toward a more mature understanding of computing as a discipline (Grover, Pea & Cooper, 2015).

Examples include:

Scratch	Snap	mBlock
		

Note

A program is a sequence of symbols that specifies a computation.

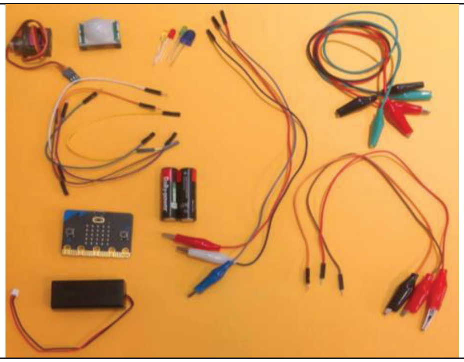
A programming language is a set of rules that specify which sequences of symbols constitute a program, and what computation the program describes.

A programming language is an abstraction mechanism. It enables a programmer to specify a computation abstractly, and to let a program (usually called an assembler, compiler or interpreter) implement the specification in the detailed form needed for execution on a computer (Ben-Ari, 2006)

Where learners struggle, physical coding or coding cards could be used as support and remediation.

2.10.2 Robotics resources

Refer to Figure 2.9 and Table 2-5, Table 2-8 and Table 2-9 for Robotics components

		
Plastic gears, fans, base plates (Lego plates) Rubber bands Fans Pully's Connectors Pipe cleaners Googley eyes Plastic wheels Screws and nuts Pipe cleaners	Simple switches Wire Straws String Cable ties Double sided tape Small pieces of wood blocks (available from arts and craft shops) Plastic bottle caps (for wheels) Servo Motor (E.g., SG90)	PIR Motion Sensor (E.g., HC-SR501) Crocodile to Male pin connectors Crocodile to Female pin connectors Crocodile clips (Clips at both ends) Batteries LEDs Connection wire with pins Foil Microcontroller
DC Motors DC Water pumps Servos 180 degrees and Continuous 360 degree	Battery packs (9V maximum) Breadboard Microcontroller breakout board Different electronic components Sensors (PIR, Moisture, Water Level, IR, Ultrasonic sensors)	Motor driver board LEGO/Building Blocks for rapid prototyping

2.10.3 Digital Concepts Resources

- Sample technologies and components (e.g., mobile phone, tablet, laptop (with screen, keyboard and mouse), etc.)
- Pictures of computing devices, input devices, output devices, networks, etc.

2.11 OVERVIEW OF SENIOR PHASE CODING AND ROBOTICS

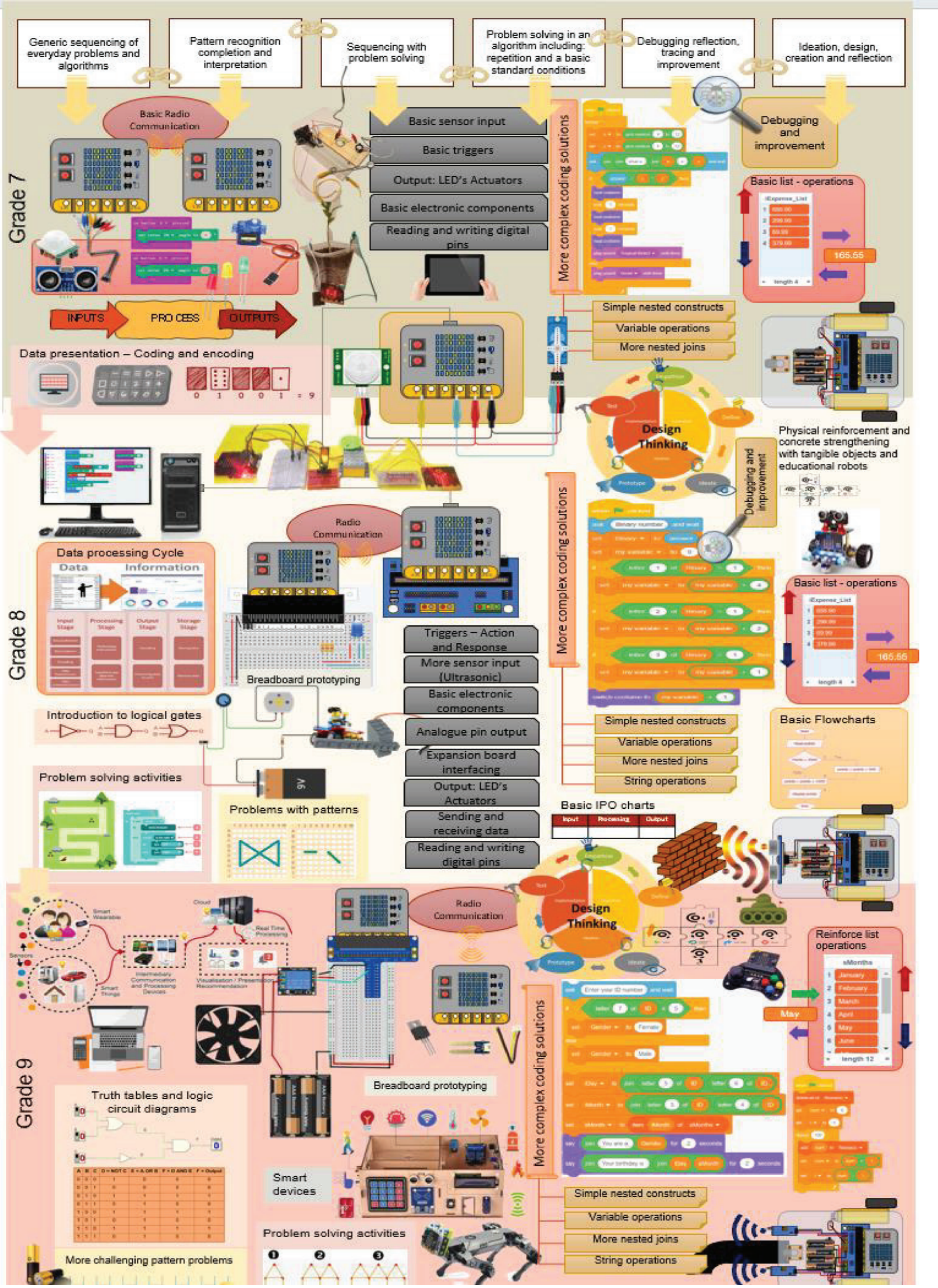


Figure 2-13 Coding & Robotics Overview Grade 7 - 9

2.12 FOCUS OF CONTENT AREAS

2.12.1 Coding

Table 2-4: Coding content focus and progression

Competency	Grade 7 (intermediate level)	Grade 8 (competent level)	Grade 9 (proficient level)
C.1 **Apply computational thinking skills to develop a set of logical instructions to solve a problem.**	Progression in problem-solving mostly lies in gradually increasing the scope and complexity of problems.		
	Using simple to moderate problems: • Develop a set of logical instructions to solve a moderate problem that includes (where appropriate): - sequences of commands - repetition - conditional constructs **Refer to Table 2-7** • Trace, evaluate, correct or complete a set of logical instructions (algorithm) (link to C.2, C.3, C.4, C.5, C.6 and C.7)	Using moderate to intermediate problems: • Develop a set of logical instructions to solve a problem that includes (where appropriate): - sequences of commands - repetition - conditional constructs **Refer to Table 2-7** • Trace, evaluate, correct or complete a set of logical instructions (algorithm) (link to C.2, C.3, C.4, C.5, C.6 and C.7)	Using intermediate to little more involved problems: • Develop a set of logical instructions to solve a problem that includes (where appropriate): - sequences of commands - repetition - conditional constructs **Refer to Table 2-7** • Trace, evaluate, correct or complete a set of logical instructions (algorithm) (link to C.2, C.3, C.4, C.5, C.6 and C.7))
	Computational thinking is infused and used in all aspects of coding/problem solving activities as follows: Use **abstraction** to simplify complex problems by reducing it to its most essential components. It helps to gain a deeper understanding of their underlying structure and develop more effective solutions. It allows you to focus on the essential aspects while minimizing distractions from irrelevant details. **Abstraction** is also used to make a visual representation of solution. Use **decomposition** to help to simplify complex problems by breaking them down into manageable parts, that allows one to address each component individually. It enables a systematic approach to problem-solving and enhances ones understanding of the problem's structure and relationships. Use **pattern recognition** to leverage existing knowledge and experiences to identify meaningful regularities in data or situations. Recognising patterns, helps to gain insights, make predictions, apply them to new problems, and develop more effective problem-solving strategies. Use **algorithmic thinking** to develop a series of precise, logical steps or instructions (algorithm) to accomplish a task or solve a problem in an organised and methodical manner.		
C.2 **Present a competent coding solution using symbolic or written statements representing sequences of commands, single repetition, and conditional constructs.**	Working with a coding solution (set of logical instructions) for simple to moderate problems: • Translate a coding solution (set of instructions/algorithm) into programming code (e.g. block-based coding instructions /Tanks cards, etc.). • Use code (symbols/ blocks/written statements to represent actions and operations to accomplish a particular task/solve a programming problem. • Group instructions/code blocks to represent repetition (or a statement indicating repetition) (Done in relation to C.1, C.4 and C.5)	Working with a coding solution (set of logical instructions) for moderate to intermediate problems: • Translate a coding solution (set of instructions/algorithm) into programming code (e.g. block-based coding instructions /Tanks cards, etc.). • Use code (symbols/ blocks/written statements to represent actions and operations to accomplish a particular task/solve a programming problem. • Group instructions/code blocks to represent repetition (or a statement indicating repetition) (Done in relation to C.1, C.4 and C.5)	Working with a coding solution (set of logical instructions) for intermediate to little more involved problems: (Done in relation to C.1) • Translate a coding solution (set of instructions/algorithm) into programming code (e.g. block-based coding instructions/Tanks cards, etc.). • Use code (symbols/ blocks/written statements to represent actions and operations to accomplish a particular task/solve a programming problem. • Group instructions/code blocks to represent repetition (or a statement indicating repetition) (Done in relation to C.1, C.4 and C.5)
C.3 **Interpret and execute a given symbolic or written set of commands**	Using simple to moderate problems: • Execute a set of commands using unplugged activities (physically, on paper, coding cards) or an educational tool (e.g. block-based coding software). Done in relation to C.1 and C.2 • Determine the output of a given algorithm (set of commands) or of given program code (e.g. trace block-based commands) to determine the output or explain what the code/program does).	Using moderate to intermediate problems: • Execute a set of commands using unplugged activities (physically, on paper, coding cards) or an educational tool (e.g. block-based coding software). Done in relation to C.1 and C.2 • Determine the output of a given algorithm (set of commands) or of given program code (e.g. trace block-based commands) to determine the output or explain what the code/program does).	Using intermediate to little more involved problems: • Execute a set of commands using unplugged activities (physically, on paper, coding cards) or an educational tool (e.g. block-based coding software). Done in relation to C.1, C.2 and C.3 • Determine the output of a given algorithm (set of commands) or of given program code (e.g. trace block-based commands) to determine the output or explain what the code/program does).
C.4 **Debug a given**	• Reinforce reading and understanding a simple to moderate problem.	• Reinforce reading and understanding a moderate to intermediate problem.	• Reinforce reading and understanding an intermediate to little more involved

symbolic or written set of instructions.	• Interpret/execute/trace a given set of commands to determine correctness of the solution/if the correct output is achieved. • Complete an incomplete a set of commands provided to solve a given moderate problem. • Inspect/trace a coding solution (set of commands) for an error or errors and correct if necessary. (Unplugged and Plugged). Done in relation to C.1, C.2, C.3 and C.4	• Interpret/execute/trace a given set of commands to determine correctness of the solution/if the correct output is achieved. • Complete an incomplete a set of commands provided to solve a given moderate problem. • Inspect/trace a coding solution (set of commands) for an error or errors and correct if necessary. (Unplugged and Plugged). Done in relation to C.1, C.2, C.3 and C.4	problem. • Interpret/execute/trace a given set of commands to determine correctness of the solution/if the correct output is achieved. • Complete an incomplete a set of commands provided to solve a given moderate problem. • Inspect/trace a coding solution (set of commands) for an error or errors and correct if necessary. (Unplugged and Plugged). Done in relation to C.1, C.2, C.3 and C.4
C.5 **Evaluate a given solution towards potential improvement.**	Using simple to moderate problems: • Reflect and report on a given solution by asking the following questions (critical thinking): - What happened? - Why has it happened? - What can be learnt? - How can the solution be improved? • Inspect a set of commands (algorithm/program) and reflect to improve it or provide a better alternative (e.g. reducing the number of steps/instructions using a loop for repetitive steps/patterns Link to C.6 and C.7.	Using moderate to intermediate problems: • Reflect and report on a given solution by asking the following questions (critical thinking): - What happened? - Why has it happened? - What can be learnt? - How can the solution be improved? • Inspect a set of commands (algorithm/program) and reflect to improve it or provide a better alternative (e.g. reducing the number of steps/instructions using a loop for repetitive steps/patterns Link to C.6 and C.7	Using intermediate to little more involved problems: • Reflect and report on a given solution by asking the following questions (critical thinking): - What happened? - Why has it happened? - What can be learnt? - How can the solution be improved? • Inspect a set of commands (algorithm/program) and reflect to improve it or provide a better alternative (e.g. reducing the number of steps/instructions using a loop for repetitive steps/patterns Link to C.6 and C.7
C.6 **Recognise and interpret patterns in symbolic sets of data or visualisations.**	• Identify a pattern (e.g. a pattern in coding instructions, numbers, symbols, blocks, characters, sequences, etc) • Interpret, explain and complete/extend a pattern (describe the pattern rule, use the pattern rule to complete/extend the pattern or make predictions) Link to C.1 and C.2	• Identify a pattern (e.g. a pattern in coding instructions, numbers, symbols, blocks, characters, sequences, etc) • Interpret, explain and complete/extend a pattern (describe the pattern rule, use the pattern rule to complete/extend the pattern or make predictions) Link to C.1 and C.2	• Identify a pattern (e.g. a pattern in coding instructions, numbers, symbols, blocks, characters, sequences, etc) • Interpret, explain and complete/extend a pattern (describe the pattern rule, use the pattern rule to complete/extend the pattern or make predictions) Link to C.1 and C.2
C.7 **Create or complete a pattern to represent a data set.**	• Complete a simple to moderate pattern that is part of a data set or programming solution. • Create a simple to moderate pattern to form part of a data set or a programming solution. • Generalise a basic pattern based on the pattern rule Done in relation to C.6 Link to C.1 and C.2	• Complete a moderate to intermediate pattern that is part of a data set or programming solution. • Create a simple to moderate pattern to form part of a data set or a programming solution. • Generalise a simple pattern based on the pattern rule. • Incorporate the generalised pattern as part of a programming solution. Done in relation to C.6 Link to C.1 and C.2	• Complete an intermediate to little more involved pattern that is part of a data set or programming solution. • Create an intermediate to little more involved pattern to form part of a data set or a programming solution. • Generalise a moderate to intermediate pattern based on the pattern rule. • Incorporate the generalised pattern as part of a programming solution. Done in relation to C.6 Link to C.1 and C.2

Note

Linked competencies can be grouped/done together within one lesson/activity where appropriate.

2.12.2 Robotics

Table 2-5: Robotics content focus and progression

<table>
<tr><th>Outcome</th><th>Grade 7
(intermediate level)</th><th>Grade 8
(competent)</th><th>Grade 9
(proficient level)</th></tr>
<tr><td>R.1
Explain what a robot is in competent terms.</td><td>Provide a basic description of a robot in terms of (extend from Grade 6):
• definition (what it is)
• purpose
• the contexts that they operate in.
• concepts regarding the relationship between the composition of a robot - basic parts (sensors, controllers, actuators, power source)
• evolution of robots / advancements of robots (also link to the concept of AI – elementary reference to automatic decisions)
Link to R.2 and R.3</td><td>Provide a definition and basic description of a robot that is used in the security and protection industry with reference to:
• Various components
• Contexts that they operate in
• Potential application of AI
• Ethical and legal implications.
Link to R.2 and R.3</td><td>Provide a definition and basic description of a robot (e.g. drone) that is used for rescue or in unsafe and inaccessible places to operate with reference to:
• Various components
• Contexts that they operate in
• Potential application of AI
• Ethical and legal implications.
Link to R.2 and R.3</td></tr>
<tr><td>R.2
Identify different types of robots.</td><td>Learners can outline different types of robots in terms of their use and application, i.e.
• Simple Task-Oriented Robots
• Chatbots (in relation to automation and virtual robots used in the information systems space)</td><td>Learners can outline different types of robots in terms of their use and application, i.e.,
• Medical Robots
• Agricultural Robots
• Virtual Customer Service Agents (in relation to automation and virtual robots used in the information systems space)</td><td>Learners can outline different types of robots in terms of their use, composition, and application, i.e.,
• Artificial Intelligence (AI) Robots
• Exoskeleton Robots
• Industrial Robots
(in relation to automation and virtual robots used in the information systems space)
• AI-Powered Virtual Assistants
• Virtual IT Support Bots</td></tr>
<tr><td>R.3
Outline the different components of a robot</td><td>Provide a moderate reference to the components of a robot or physical computing artefact and their purpose (refer to section 2.16.3))
Link to R.1, R.2 and R.4 – R.7</td><td>Provide a moderate to intermediate reference to the components of a robot or physical computing artefact and their purpose (refer to section 2.16.3))
Link to R.1, R.2 and R.4 – R.7</td><td>Provide an intermediate reference to the components of a robot or physical computing artefact and their purpose (refer to section 2.16.3))
Link to R.1, R.2 and R.4 – R.7</td></tr>
<tr><td>R.4
Present an understanding of how robots affect the world.</td><td>• Differentiate between various robots, such as industrial robots, service robots, and educational robots
• Describe the positive and negative impacts of robots on human lives.
• Discuss the advantages and disadvantages of using robots, including increased productivity, convenience, and potential job displacement.
• Design a simple task for a hypothetical robot and present it to the class.
Link to R.1 – R3 and R.5 - R.7</td><td>• Present a basic understanding and consider the potential ethical dilemmas related to robotic technologies, such as privacy concerns, AI bias, and the responsibility for actions performed by autonomous robots.
• Compare and contrast the impact of robots on the economy in different countries.
• Present a basic discussion on how different nations are adapting to the robotic revolution, considering economic implications automation.
Link to R.1 – R3 and R.5 - R.7</td><td>• Explain the possibilities of advanced robots with artificial intelligence, considering their impact on employment, social structures, and human interaction.
• Discuss the advantages and disadvantages of utilizing robots in dangerous situations and space exploration.
• Outline the concerns and potential issues related to address concerns related to robot-human interactions, employment shifts, and privacy concerns, promoting responsible and ethical implementation in the 4IR+ society.
Link to R.1 – R3 and R.5 - R.7</td></tr>
<tr><td rowspan="3">R.5
Design a competent artefact based on a set of design specifications.</td><td>• Design simple to moderate robot artefacts using the design thinking process.
Refer to 2.16.1, 2.16.1 and 2.16.3 for content and progression.
Link to R.1 – R.4 and R.6, R.7</td><td>• Design moderate to intermediate robot artefacts using the design thinking process/
Refer to 2.16.1, 2/16.1 and 2.16.3 for content and progression
Link to R.1 – R.4 and R.6, R.7</td><td>• Design intermediate to little more involved robot artefacts using the design thinking process.
Refer to 2.16.1, 2/16.1 and 2.16.3 for content and progression
Link to R.1 – R.4 and R.6, R.7</td></tr>
<tr><td colspan="3">Design thinking is infused and used when creating robotic artifacts as follows:
Empathise: Ask questions to find out what the problem is and to identify challenges related to the problem as well as to identify ways to solve the challenges
Define: Specify the detail of the problem
Ideate: Imagine and brainstorm different ideas for solving the problem and choose the best idea
Prototype (Plan and design): Draw a competent picture (abstraction) and write down the material you will need. Then write down step-by-step instructions (algorithm) for implementing the idea.
Test (Create/implement, test, reflect and improve): Follow the design (picture) and plan (algorithm) and build the artefact. Then test it to see if it works and correct/improve where necessary.</td></tr>
<tr><td colspan="3">The progression mostly lies in the gradual increase in scope and complexity of artefacts.</td></tr>
</table>

R.6 **Mimic the operations of a robot**	Use a ***simulated*** block-based environment with a microcontroller to mimic the operations of a robot: • Align coding concepts to be used with the coding concepts covered and mastered in the coding section. • Includes role play (acting out), and tangible activities. • Includes the use of appropriate paper-based exercises. Link to R.3, R.5 and R.7 In addition, schools can also opt to use tangible tools and educational robots (OPTIONAL) for reinforcement.	Use a ***simulated*** block-based environment with a microcontroller or to mimic the operations of a robot: • Align coding concepts to be used with the coding concepts covered and mastered in the coding section. • Includes role play (acting out), and tangible activities. • Includes the use of appropriate paper-based exercises. Link to R.3, R.5 and R.7 In addition, schools can also opt to use tangible tools and educational robots (OPTIONAL) for reinforcement.	Use a ***simulated*** block-based environment with a microcontroller to mimic the operations of a robot: • Align coding concepts to be used with the coding concepts covered and mastered in the coding section. • Includes role play (acting out), and tangible activities. • Includes the use of appropriate paper-based exercises. Link to R.3, R.5 and R.7 In addition, schools can also opt to use tangible tools and educational robots (OPTIONAL) for reinforcement.
	The scope and complexity are gradually increased in relation to the coding features, operations and structures (code blocks/coding constructs – coding knowledge and skills) as well as the physical features and components introduced per term per year as well as in terms of the complexity of the problem. **Refer to Section 2.16.1, 2.16.1.2 and 2.16.3 for content/skill per grade and for progression**		
R.7 **Create, test, and execute a set of robotic instructions.**	Using simple to moderate problems: • Develop solutions to solve specific problems • Translate solution instructions (algorithms) into code (using virtual robots with instructions in a tangible or non-tangible coding environment (software) or using physical educational robotic tools or both). • Implement, test, modify and/or improve solutions. (Link to C.1 to C.5 as well as R.5 to R.6)	Using moderate to intermediate problems: • Develop solutions to solve a specific problem. • Translate solution instructions (algorithms) into code (using virtual robots with instructions in a tangible or non-tangible coding environment (software) or using physical educational robotic tools or both). • Implement, test, modify and/or improve solutions. (Link to C.1 to C.5 as well as R.5 to R.6)	Using intermediate to little more involved problems: • Develop solutions to solve a specific problem. • Translate solution instructions (algorithms) into code (using virtual robots with instructions in a tangible or non-tangible coding environment (software) or using physical educational robotic tools or both). • Implement, test, modify and/or improve solutions. (Link to C.1 to C.5 as well as R.5 to R.6)
	The scope and complexity are gradually increased in relation to the coding features, operations and structures (code blocks/coding constructs) as well as physical features and components introduced per term per year as well as in terms of the complexity of the problem. **Refer to Section 2.16 for content/skill per grade and for progression**		

Note

Linked competencies can be grouped/done together within one lesson/activity where appropriate.

Note

Processes such as computational thinking and design thinking, is like riding a bicycle/driving a car. With repeated practise and experience, they become intrinsic. It means that the thinking process or skills required to perform the activity become so well-practiced and internalised that they become automatic and almost second nature.

As with driving a car repeatedly, you gain experience, become able to recognise patterns on the road, formulating driving “algorithms” (decomposing driving tasks, breaking distance calculations, etc.). With enough practise, decision-making when driving becomes automated (e.g. using your turn indicators). Over time you become more skilled at abstracting relevant information when driving, and as you encounter different driving scenarios, you learn from your mistakes and “debug” your driving skills. Also, with experience, you become skilled at processing multiple factors simultaneously (e.g., consider traffic signals, the behaviour of other drivers, while maintaining your focus on the road ahead). Eventually, you go into unconscious competence mode – performing these actions almost involuntarily.

These thinking processes therefore become intrinsic, and the problem-solving approaches and mental processes involved become an integral part of your behaviour. At first though, they must be consciously learned and repeatedly practised.

2.12.3 Digital Concepts

Table 2-6: Digital Concepts content focus and progression

Competency	Grade 7	Grade 8	Grade 6
D.1 **Outline the concept of technology and purpose of information technology (IT).**	• Explain what technology and information technology is. • Distinguish between technology and information technology. • Relate the concept information technology to that of a tool such as the use of computers and software or other computing devices and tools such as smart watches, etc. to process large amounts of data rapidly, to provide information that enables informed decisions. • Know the role of IT and IT professionals (programmers) as well as the role mathematics, statistics and simulations play to analyse data and information, make predictions, and guide decisions. • Identify the use of information technology in a specific real-world scenario and explain the purpose. • Identify examples of information technology and relate their use and purpose to everyday life. Link to D.3, D.4 and D.5	• Explain what a computer/computing device in the context of information technology is. • Relate the concept of technology and information technology to that of a tool such as the use of computers and software (applications and concept of operating system) to manage and process data into information to enable informed decisions. • Understand Know the role of IT and IT professionals (programmers) as well as the role mathematics, statistics and simulations play to analyse data and information, make predictions, and guide decisions. • List examples of computers and computing devices and relate their use and purpose to everyday life. • Understand the purpose of information technology and its role in general. • Identify the information technology used in a specific real-world scenario (e.g. entertainment, shopping) and explain the purpose. Link to D.3, D.4 and D.5	• Explain what a computer/computing device in the context of information technology is. • Relate the concept of computers and microcontrollers to that of an ICT tool. • Describe examples of computers and microcontrollers and relate their use and purpose to everyday life. • Describe networks as a key component of technology, representing interconnected systems that facilitate the transmission of data and communication between devices or users. • Relate technology and IT to the Internet of Things (IoT) that uses technologies, IT and networks to collect and exchange data over the internet or other communication networks. • Compare and evaluate the role of information technology in two different contexts (e.g. education, shopping, and entertainment) and discuss advantages and disadvantages. Link to D.3, D.4 and D.5 and Rs
D.2 **Recognise that he or she is living as citizens in a digital world.**	• Provide an explanation of the digital world and a digital citizen • Explain and demonstrate the responsible use technology and computers in the classroom. • Recognise the dangers of the online environment (online predators, addiction, and distraction). • Provide an understanding of - Evaluating information (false/fake news), sources. - Cyberbullying and how to deal with it. - Reason for using passwords/ pins (security). - The concept and dangers of sharing information like personal information, usernames, and passwords. - A digital footprint - Social responsibility Link to D.6	• Provide an explanation of the digital world and a digital citizen • Understand how to use technology and computers in the classroom responsibly and when to report unsuitable use, unauthorised access of content and/or contact. • Understand the dangers of the online environment (online predators, addiction, false information) • Understand the impact in terms off - Cyberbullying and how to deal with it. - Reason for using passwords/ pins (security). - The concept and dangers of sharing information like personal information, usernames, and passwords/pins. - A digital footprint Link to D.6	• Explain what the digital world and a digital citizen is. • Explain how to use technology and computers in the classroom responsibly and when to report unsuitable use, unauthorised access of content and/or contact. • Explain ethical issues and dangers associated with the use of information technology, including privacy, security, copyright, false information and inappropriate content. • Provide guidelines on how to manage: - Cybersecurity, including phishing and ransomware - Cyberbullying - Passwords/pins (security). - Sharing of personal information. - Digital footprints Link to D.6
D.3 **Demonstrate an understanding of the concept of a computing device.**	• Know what a computing device in general is. • Know the parts and function of a computing device, including concepts of input, processing, output, storage and software. • Provide examples of computing devices and their main purpose • Understand the concepts of hardware and software ("apps"). • Distinguish between computers and robots. (When is a robot a computer? When is a computer a robot?) • Provide a definition and function of a modem and router as network devices	• Know the definition and function of a microprocessor and microcontroller. • Distinguish between a microprocessor and a microcontroller in terms of purpose, input/output, memory, examples and common applications. • Describe examples of common computing devices used in a network (including modem and router) and describe what they are used for/their main purpose. (Link to D.1 and D.2)	• Explain what a computing device, in general, is. • Describe the microcontroller as a computing device and it use as a component in the IoT. • Explain the purpose and role of hardware (as input, processing, storage, and output) and software as a list of instructions (apps and operating system) that the computer can follow. • Describe common computing devices such as a computer and a microcontroller and describe their input, output, processing and storage devices.

			• Describe the purpose and role of basic network devices (modem, router, firewall, IoT gateway) • Identify the instructions/software (‘apps’) one can use on the devices and the function/purpose of those (e.g., block-based coding app to write computer programs).
D.4 **Identify the common uses of ICT in the real world.**	• Define ICT (inclusion of the concept of ‘communication’ in ‘IT’ that allows people to interact in the digital world). • Identify everyday uses of ICTs, e.g., the Internet and the World Wide Web (WWW) • Understand what the Internet is • Understand what the World Wide Web (WWW) is • Distinguish between the Internet and the WWW • Understand what is required to establish a basic connection to the internet (modem, examples of different connectivity methods in a real-world scenario (Wi-Fi, cellular, cabled)) • Know that data is processed to provide information that informs decisions (Link to D.1, D.2, D.3 and D.4)	• Describe ICT (inclusion of ‘communication’ in ‘IT’ that allows people to interact in the digital world). • Describe everyday uses of ICTs, in terms of a basic home network connected to the Internet and devices/technology required (modem, router, firewall, ISP) and examples of different connectivity methods (Bluetooth, Wi-Fi, cellular, cabled). • Understand what a network is, the purpose of a network and list basic advantages of a network • List different uses of ICTs e.g. school network / entertainment / shopping) (Link to D.1, D.2, D.3 and D.4)	• Explain ICT as an umbrella term that includes communication devices and systems. • Explain everyday uses of ICTs in terms of a basic home network connected to the Internet; (basic devices/technology required (modem and router, firewall, ISP) and connectivity methods (Bluetooth, Wi-Fi, cellular, cabled). • Describe the Internet of Things (IoT) as a network of interconnected physical devices or "things" that are embedded with sensors, software, and other technologies to collect and exchange data over the internet or other communication networks. (Link to D.1, D.2, D.3 and D.4)
D.5 **Differentiate between the components of an ICT system.**	• Know what an ICT system is (includes hardware (including computing devices such as modem router), software, data, communication (connectivity such as Wi-Fi, cellular, cabled), networks and people, (e.g. users)). • Provide an example of a basic ICT system. (Link to D.1, D.2, D.3 and D.4)	• Describe an ICT system (basic home/school network and components) in terms of hardware (various computing devices), software and data, communication (modem, router) connectivity such as Wi-Fi, cellular, cabled) – including security (firewall), ISP, cloud storage service). • Provide common examples of ICT systems with basic descriptions (education, agriculture, media and entertainment) (Link to D.1, D.2, D.3 and D.4)	• Explain an ICT system (includes hardware & software (computing devices with software such as computers, modem, router, firewall), data, communication (network and Internet), connectivity (Wi-Fi, Bluetooth, cellular, cabled) and people) (modem, router, firewall), peripherals, e.g. printer, ISP) • Identify the key components of an IoT system including microcontroller sensors, actuators, physical objects. • Cloud service (storage, software) (Link to D.1, D.2, D.3 and D.4)
D.6 **Explain how the adaptation of technology impacted the world we work and live in.**	• Understand how technology impacts how we manage data and information and interact with others. • Understand what cloud storage is and that it is a service that allows one to store one’s files online, • Relate the concept of cloud storage to a network (internet) (Link to D.2, D3, D.4 and D.5).	• Understand the impact of networks, their role in communication, society. • Understand what cybercrime is and what it includes - Steal or alter data - Steal personal data/information such as passwords - Gain unlawful use of computers or services or access to networks - Create and distribute malware • Describe how cybercrime impact on the world we live in • Understand and describe how technology impacts the following: - Communication - Transportation - Education - Information and information sources (False information/Fake news) • Evaluate information and information sources • Describe the role of the WWW and networks in facilitating communication, collaboration, and information sharing (Link to D.2, D3, D.4 and D.5).	• Understand and explain of how technology impacts the following: - Interaction with others (global connectivity) - Communication and collaboration - Access to information (including False information/Fake news) - Entertainment and media (movie/audio streams, music instruments, games) - Education and e-Learning - Smart technologies - Healthcare - Agriculture - Transportation - Cybersecurity challenges - Privacy concerns • Understand how the IoT impacts our lives and provide examples • Evaluate information and information sources (including fact checking – link to fake news)) (Link to D.2, D3, D.4 and D.5).

D.7 **Present a basic understanding of the concept of input processing and output.**	• Understand that input results in a form of output through processing – also link to IPO table in coding. • Understand that different forms of input result in different actions/outputs. (e.g., use of sensors and actuators or user input and related output) • Distinguish between input through instructions that are executed and results in action and output as a form of communication from the device. • Relate the concept of input, processing and output to a network. • Understand the concept of cloud storage. (Link to D.3, D.4, D.10, C.2, R.6, R.7)	• Describe the concept of input that results in a form of output because of processing using examples. • Describe different forms of input and output because of processing • Identify output as a form of communication from the device. • Describe the information processing cycle regarding a given real-life scenario (input, processing, output, storage, communication) • Describe the concept of cloud storage. • Describe the concept of GIGO (that the incorrect input will result in incorrect output). (Link to D.3, D.4, D.10, C.2, R.6, R.7)	• Explain how different forms of input result in different forms of output because of processing. • Explain input, processing and output with regards to microcontrollers. • Explain the interaction/relationship between input, processing, and output (e.g. when coding and using a microcontroller with sensors and actuators). • Differentiate between local and cloud storage. • Understand that incorrect input results in incorrect output (GIGO) (Link to D.3, D.4, D.10, C.2, R.6, R.7)
D.8 **Interpret a pattern to represent or communicate a message or image.**	• Interpret a simple pattern (e.g., representations such as morse code, binary code, a simple cipher to communicate (encode) a message. • Interpret an image (e.g. a 'road sign' or symbolic representations such as smileys) • Decrypt a message. (Link to D.9 and C.1, C.2)	• Interpret a simple pattern (e.g., representations such as morse code, binary code, a simple cipher) to communicate (encode) a message. • Interpret an image (e.g. a 'road sign' or symbolic representations such as smileys) • Decrypt a message. (Link to D.9 and C.1, C.2)	• Interpret a simple pattern (e.g., representations such as morse code, binary code, a simple cipher) to communicate (encode) a message. • Interpret an image (e.g. a 'road sign' or symbolic representations such as smileys) • Decrypt at messages. (Link to D.9 and C.1, C.2)
D.9 **Create a pattern to represent or communicate a message or image.**	• Create a pattern to communicate a message (e.g., use a simple cipher to encrypt/decrypt and communicate a message) or design an image (e.g., text or LEDs to communicate a message). • Use binary code to communicate a message. • Simulate/display a message/ game (e.g., scrolling 'billboard message' or rock, paper, scissors game) on a microcontroller (LEDs on grid). (Link to D.8, C.1, C.2 R.5, R.6, R.7)	• Create a pattern to communicate a message (e.g., use a simple cipher to encrypt/decrypt and communicate a message) or design an image (e.g., text or LEDs to communicate a message). • Use binary code to communicate message. • Simulate/display the message (e.g., symbols such as a smiley or heart) on a microcontroller (LEDs on grid). (Link to D.8, C.1, C.2 R.5, R.6, R.7)	• Create a pattern to communicate a message (e.g., use a simple cipher to create (encrypt) and communicate a message or design an image (e.g., text or LEDs to communicate a message) • Convert ASCII code to binary to communicate a message. • Simulate/display a message/ game (e.g., scrolling 'billboard message' or rock, paper, scissors game) on a microcontroller (LEDs on grid). (Link to D.8, C.1, C.2 R.5, R.6, R.7)
D.10 **Demonstrate a basic proficiency in the application of digital skills.**	• Use a block-based application. • Explain and demonstrate file and folder management using an appropriate folder hierarchy, file naming conventions and extensions. • Explain the purpose of a file name and extension. • Save and Open filles from within an application as well as following a file path. • Fluent use of different input and output devices to perform tasks and functions. • Design a sprite and a backdrop to import and use in a block-based application, using an application such as Paint (link to D.9). • Design a customised 'GUI' and sprites for a block-based application. (Link to C.2 – C.5 and R.5 – R.7)	• Use a block-based application. • Explain and demonstrate file and folder management using an appropriate folder hierarchy, file naming conventions and extensions. • Explain the purpose of a file name and extension. • Save and Open filles from within an application as well as following a file path. • Fluent use of different input and output devices to perform tasks and functions. • Find suitable images saved on the computer to customise and import as a sprite or a backdrop for use in a block-based application (link to D.9). • Introduction to, and elementary use of a presentation application such as PowerPoint. (Link to C.2 – C.5 and R.5 – R.7)	• Explain and demonstrate the concept of working in and navigating an application (app) (Link to C.2) • Explain and competently demonstrate file and older management using an appropriate folder hierarchy, file naming conventions and extensions. • Explain the purpose of a file extension. • Save and Open filles from within an application as well as following a file path. • Fluent use of different input and output devices to perform tasks and functions. • Find suitable images on the internet to import and customise as a sprite or a backdrop to import and use in a block-based application. (link to D.9). • Basic use of a presentation application such as PowerPoint. (Link to C.2 – C.5 and R.5 – R.7)

Note

Linked competencies can be grouped/done together within one lesson/activity where appropriate.

2.13 ENVISAGED LEARNER

The Coding and Robotics learner shows an interest in technology and its application in the world. The learner can think logically and critically and is able to solve problems. Furthermore, the learner is creative and innovative as well as disciplined, focused, and persistent. The learner can also work well with others to achieve a common goal.

2.14 ARTICULATION

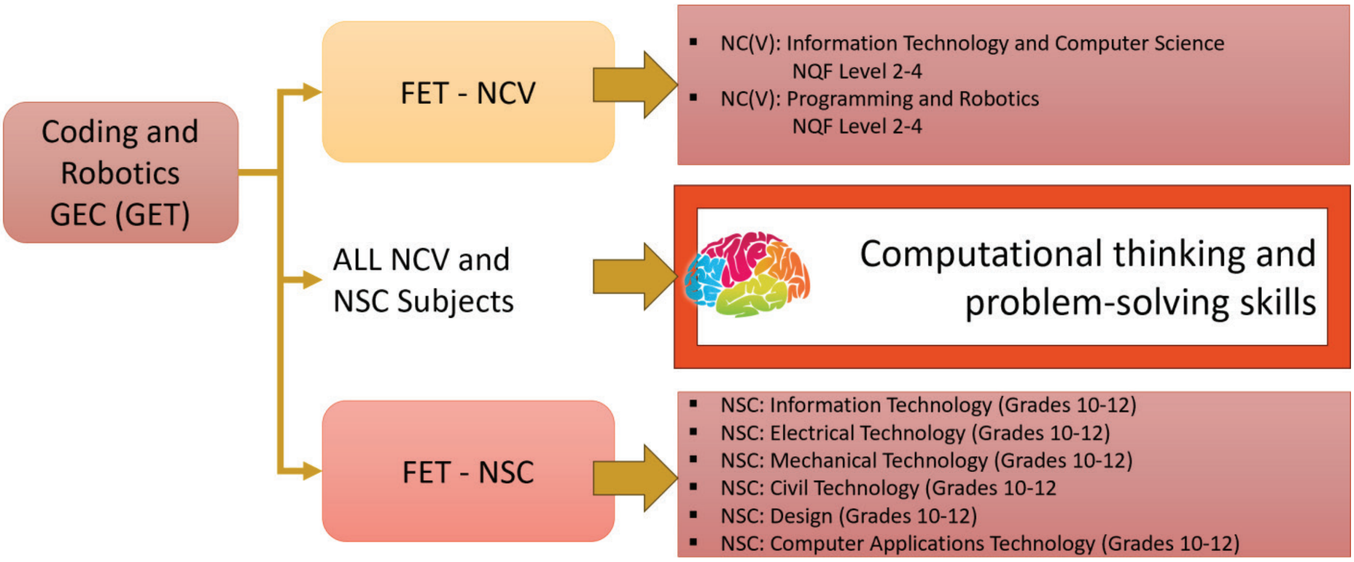


Figure 2-14 Articulation

2.15 CAREER OPPORTUNITIES

Today, digital technologies are integrated in all aspects of our lives. Digital competencies such as Coding and Robotics skills make one more employable and effective in any job and support further studies.

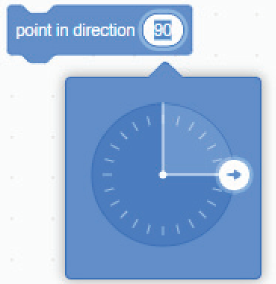
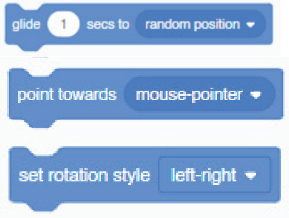
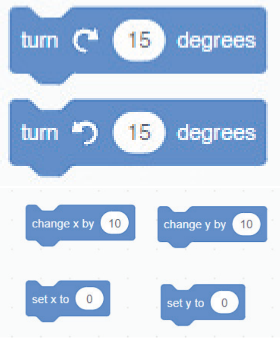
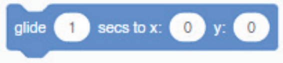
The growing ubiquity of digital technologies and the developments around the Internet of Things (IoT), automation and artificial intelligence (AI) have seen the inclusion of skills such a computational thinking, design thinking, software development (coding) and robotics in every sector of employment and entrepreneurship. Therefore, Coding and Robotics aims to equip learners with knowledge and skills that will allow them to thrive in any career and specifically in careers such as software development, robotics engineering, artificial intelligence, etc.

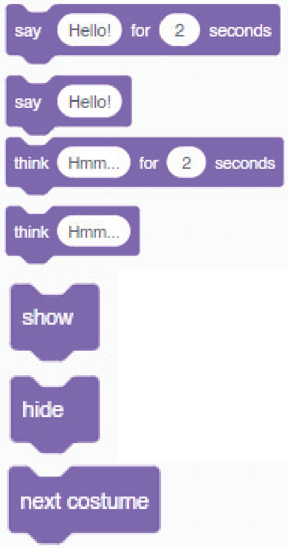
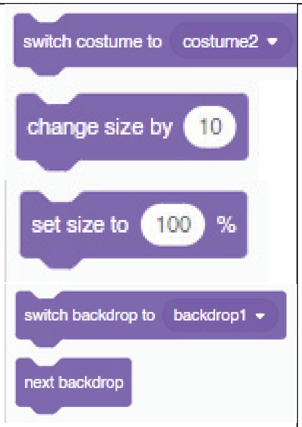
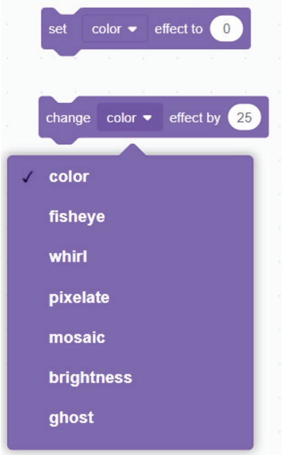
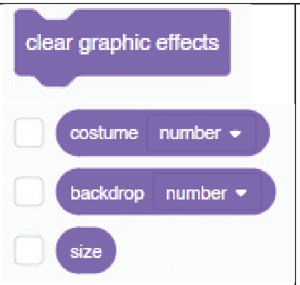
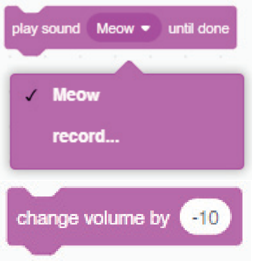
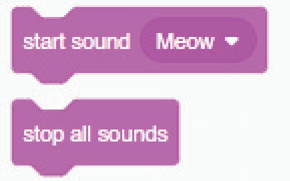
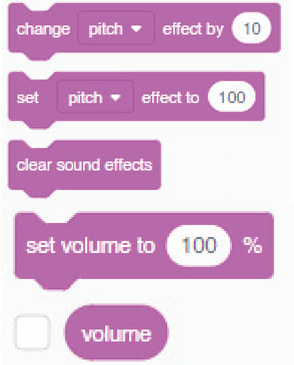
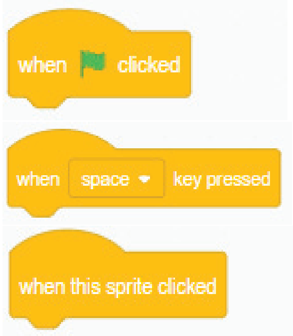
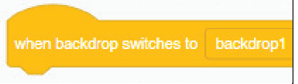
2.16 PROGRESSION AND EXIT SKILLS PER GRADE OF FOCUS AREAS

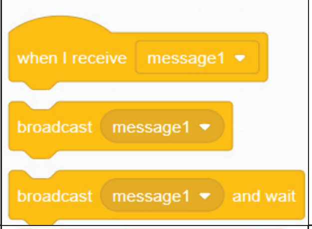
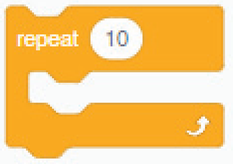
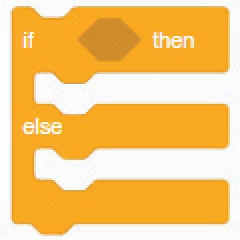
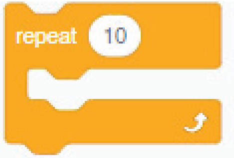
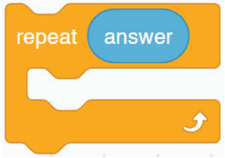
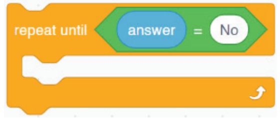
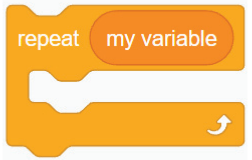
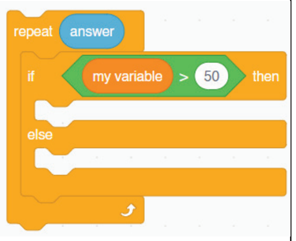
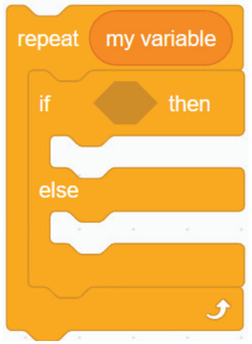
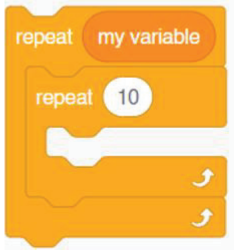
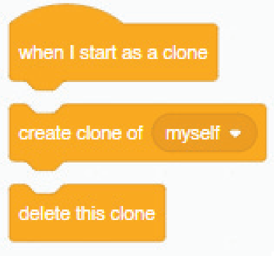
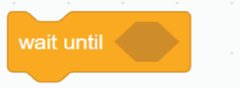
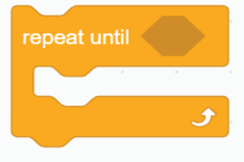
2.16.1 Coding

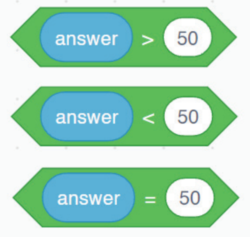
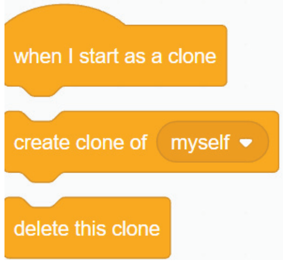
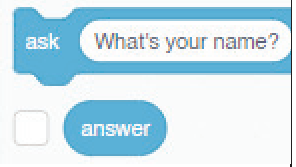
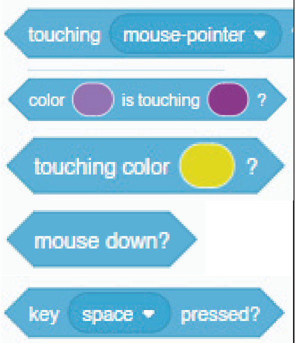
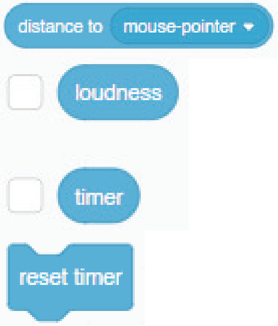
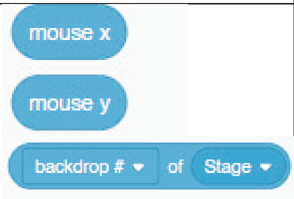
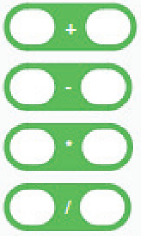
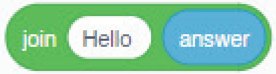
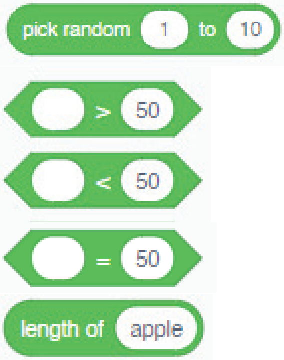
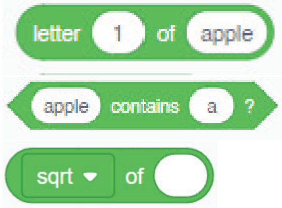
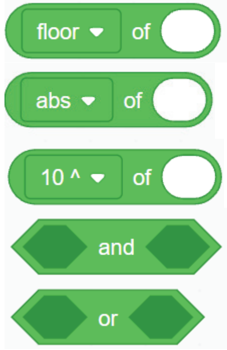
The following table provides the coding competencies that learners must demonstrate by the end of each Grade from Intermediate to Senior Phase:

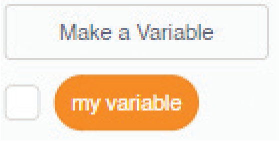
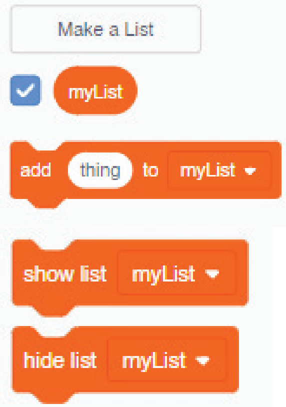
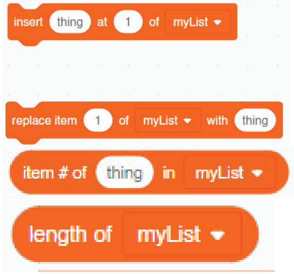
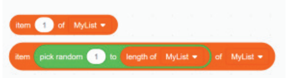
Table 2-7 Senior phase coding concepts, content and skills breakdown and progression per grade

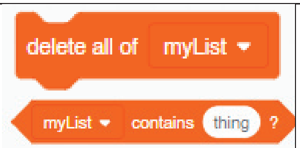
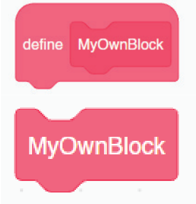
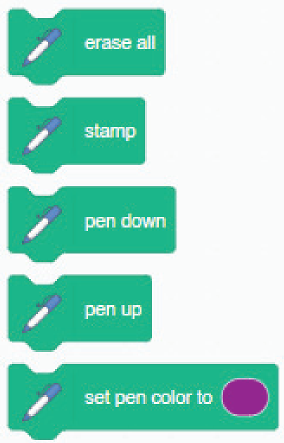
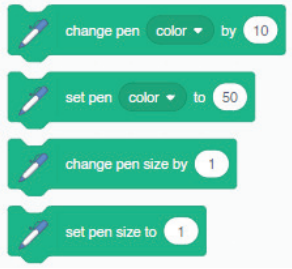
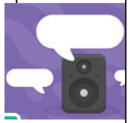
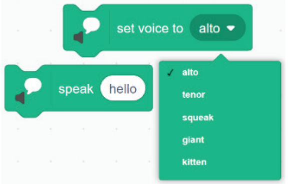
	Scratch for Intermediate and Senior phase (content breakdown and concept progression)					
	Grade 4	**+ Grade 5**	**+ Grade 6**	**+ Grade 7**	**+ Grade 8**	**+ Grade 9**
 Motion	 move 10 steps  point in direction 90  go to x: 0 y: 0 Only 0,0 for middle and to fixed given starting coordinates.  go to random position  if on edge, bounce	 glide 1 secs to random position point towards mouse-pointer set rotation style left-right	 turn ↻ 15 degrees turn ↺ 15 degrees change x by 10 change y by 10 set x to 0 set y to 0	 glide 1 secs to x: 0 y: 0  ☐ x position ☐ y position ☐ direction		

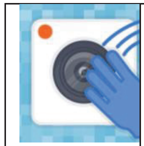
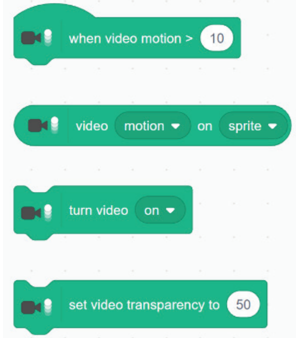
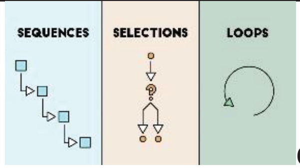
Looks						
Sound						
Events						

						
Control	  Single command. From term 3 onwards a maximum of two related commands can be grouped. E.g., (Move and Point in direction, Move and stamp, Move and wait) etc. If touching colour to continue Grade 3 knowledge  	  If with answer answer or (key pressed, e.g., space)   If condition + Relational operators with answer block, e.g.,	 With answer as a left side value answer  	  Fixed counter loop with a single If...then...else allowed	  	   

	 Single if with answer block as extended opportunity for highflyers (Optional).  If with answer				Double nested loop with guidance in problem statement where one loop is a fixed counter loop. NO additional nesting conditions. 	Programs should not require more than two nested structures Double nested loops allowed with guidance in the problem statement.
Sensing						
Operators	 Fixed values e.g., 5 + 6 		 Stacking of blocks (basic concept mastered in Grade 5) - To nest commands.			

	 join Hello banana Space manually typed after first word.	Stacking of blocks (Nesting commands)  join join join apple banana banana banana	 + *			
Variables			 Make a Variable my variable Term 2 onwards (Only) Only two variables per application (basic) with guidance  set my variable to 0 change my variable by 1 show variable my variable hide variable my variable	Limited to three – four variables in the same problem with some guidance in the problem statement	Multiple variables may be introduced with some guidance in the problem statement	
Lists				Create a simple list. Guide the learners with instructions towards its implementation in the solution  Make a List ☑ myList add thing to myList show list myList hide list myList	 insert thing at 1 of myList replace item 1 of myList with thing item # of thing in myList length of myList ** Display random ITEM from a list  item 1 of MyList item pick random 1 to length of MyList of MyList	

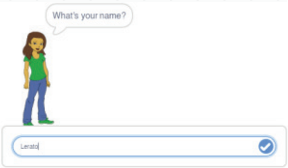
						
User defined Blocks						
Pen 						
Text to speech 		 These blocks and functions can be used to illustrate the concept of AI to the learners				
Video sensing			Video Sensing – Optional for highflyer			

			 when video motion > 10 video motion ▾ on sprite ▾ turn video on ▾ set video transparency to 50			
Application skills (IDE)	Select your own sprite Select your own background Save your program Open an existing program (Change a given sprite)	Import a picture as a sprite Import an animated gif as a sprite with costumes Import a background				
		**Concepts, constructs and practices**				
	▪ Simple sequential algorithms ▪ Everyday scenarios ▪ Sequences for integration with other subjects (e.g., Languages) + Songs ▪ Iteration on one single command for a fixed number of times ▪ Singular condition (with answer (input value) as reference comparison ▪ Fixed value calculations	▪ Simple sequential algorithms ▪ Everyday scenarios ▪ Sequences for integration with other subjects (e.g., Languages) + Songs ▪ Change the costume of a sprite ▪ Forever loops ▪ Singular condition with else (with answer (input value) as reference comparison ▪ Forever loop + 1 (nested conditional structure)	▪ Simple sequential algorithms ▪ Everyday scenarios ▪ Sequences for integration with other subjects (e.g. Languages) + Songs + Modelling a traffic light. ▪ Change the costume of a sprite ▪ Use of iteration (simple) with variable condition ▪ Change the backdrop ▪ Draw shapes with loops ▪ Forever loops ▪ Loop + Singular nested conditional structure ▪ Stacking (nesting of blocks) for calculations string output. ▪ Guide the learners on the use and implementation of variables	▪ Sequential algorithms ▪ Everyday scenarios with simple problems ▪ Sequences for integration with other subjects (e.g., Languages) + Songs + Technology + Mathematics, Natural Sciences, Life orientation. ▪ Change the costume of a sprite based on a condition or broadcast ▪ Use of iteration (simple) with variable condition ▪ Fixed counter outer loop, with nested simple conditions ▪ Change the backdrop ▪ Draw shapes with loops ▪ Forever loops ▪ Loop + Singular nested conditional structure ▪ Stacking (nesting of blocks) for calculations string output. ▪ Guide the learners on the	▪ Sequential algorithms ▪ Everyday scenarios with simple problems and integration with other subjects with strengthening of concepts using other subject domains. E.g. Smart plant watering system. ▪ Double nested loops with guidance ▪ Sequences for integration with other subjects (e.g. Languages) + Songs + Technology + Mathematics, Natural Sciences, Life orientation. ▪ Change the costume of a sprite based on a condition or broadcast. ▪ Add multiple sprites to a solution including stamping images. ▪ Use of iteration (simple) with variable condition	▪ Sequential algorithms ▪ Everyday scenarios with simple problems and integration with other subjects with strengthening of concepts using other subject domains. E.g., Smart plant watering system. ▪ Double nested loops with guidance ▪ Sequences for integration with other subjects (e.g., Languages) + Songs + Technology + Mathematics, Natural Sciences, Life orientation. ▪ Change the costume of a sprite. ▪ Add multiple sprites to a solution including stamping images. ▪ Use of iteration (simple) with variable condition ▪ Fixed counter outer loop, with

				use and implementation of variables. ▪ Introduce the basic concept of a list (A list should be given as part of a partial or incomplete solution) ▪ Select and display random items from a list (Concept of a list to store and use items, e.g., Values for display)	▪ Fixed counter outer loop, with nested simple conditions ▪ Change the backdrop based on a condition or broadcast. ▪ Draw more integrate shapes with loops based on user input. ▪ Forever loops ▪ Loop + Singular nested conditional structure ▪ Stacking (nesting of blocks) for calculations string output. ▪ Guide the learners on the use and implementation of variables and the development of more complex solutions. ▪ Select and display random items from a list (Concept of a list to store and use items, e.g., Values for display) ▪ Create a simple list	nested simple conditions ▪ Change the backdrop based on a condition or broadcast. ▪ Draw shapes with loops based on user input. ▪ Forever loops ▪ Loop + Singular nested conditional structure ▪ Stacking (nesting of blocks) for calculations string output. ▪ Guide the learners on the use and implementation of variables. ▪ Perform basic operations on lists. ▪ Introduce the concept of procedures through the implementation of lists. ▪ Access and modify an individual element in a list. ▪ Add and delete elements in a list. ▪ Solve more complex problems with guidance in the problem statement.
Use of sprites	One sprite only.	A maximum of two sprites.	A maximum of two sprites.	Maximum of three sprites	As required by the problem	As required by the problem
	Example - Simple sequential block  when [flag] clicked say Hello! for 2 seconds say I am going to teach you to make a Jam Sandwich for 2 se say 1) Take two bread slices. for 2 seconds say 2) Spread jam on one side of each slice. for 2 seconds say 3) Press slices together. for 2 seconds say 4) Optional: Trim crust edges. for 2 seconds say 5) Cut sandwich in half or into smaller pieces. for 2 second say 6) Enjoy your jam sandwich! for 2 seconds next costume	***Example with IF answer block***  when [flag] clicked ask Please enter the activation code and wait if answer = ABC123 then say Beep Beep - Robot activated! for 2 seconds	**Example with a loop and singular nested conditional structure** NOTE: In this activity the learners should be instructed and guided towards the use of the variables. The example below can be presented using scaffolding. - Program with two fixed numbers and the answer is checked. - Program with two variables and the answer is checked. - Program including a fixed loop for (e.g. 5 questions) - Adding a variable to count the correct answers.			

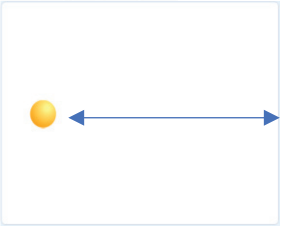
Example – Sequential (Input Output)

```
when 🏴 clicked
say Hello! My name is Abbey! for 2 seconds
ask What's your name? and wait
say join Hello answer for 2 seconds
say Good to meet you! for 2 seconds
next costume
```



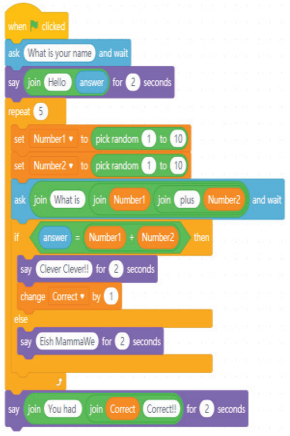
Example - If on Edge Bounce

```
when 🏴 clicked
forever
    move 10 steps
    if on edge, bounce
```



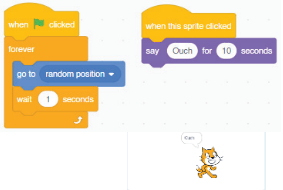
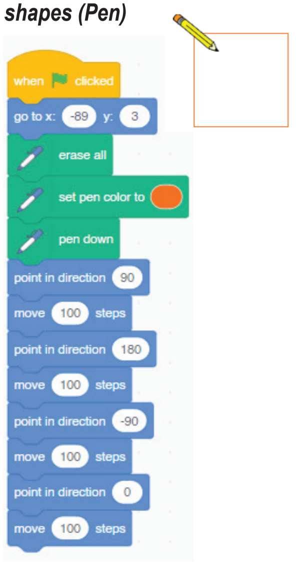
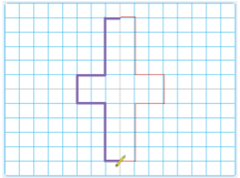
Example with repeats

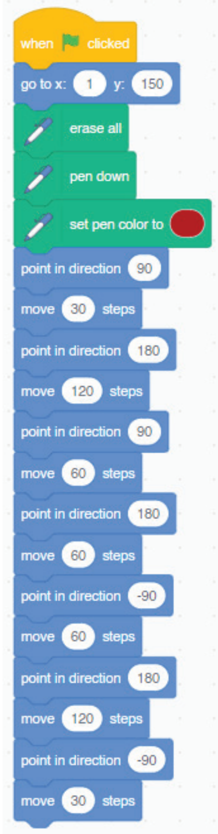
```
when 🏴 clicked
erase all
wait 1 seconds
repeat 2
    stamp
    glide 1 secs to random position ▾
wait 1 seconds
repeat 2
    say Three blind mice, for 2 seconds
    next costume
repeat 2
    say see how they run! for 2 seconds
    next costume
say They all ran after the farmer's wife; for 2 secon
say She cut off their tails with a carving knife. for 2
say Did you ever see such a sight in your life, for 2
say As three blind mice? for 2 seconds
erase all
```



```
when 🏴 clicked
ask What is your name and wait
say join Hello answer for 2 seconds
repeat 5
    set Number1 ▾ to pick random 1 to 10
    set Number2 ▾ to pick random 1 to 10
    ask join What is join Number1 join plus Number2 and wait
    if answer = Number1 + Number2 then
        say Clever Clever!! for 2 seconds
        change Correct ▾ by 1
    else
        say Eish MammaWe for 2 seconds
say join You had join Correct Correct!! for 2 seconds
```

Learners are guided with the use of variables.

	Example – Fly around (Two events click)  when clicked forever go to random position wait 1 seconds when this sprite clicked say Ouch for 10 seconds ***Example – Draw simple shapes (Pen)***  when clicked go to x: -89 y: 3 erase all set pen color to pen down point in direction 90 move 100 steps point in direction 180 move 100 steps point in direction -90 move 100 steps point in direction 0 move 100 steps ***Shapes pen – Symmetry*** Learners are provided with a stage, with a partial geometrical shape. The learners must then code how to draw the corresponding symmetrical shape. 					

	 when clicked go to x: 1 y: 150 erase all pen down set pen color to point in direction 90 move 30 steps point in direction 180 move 120 steps point in direction 90 move 60 steps point in direction 180 move 60 steps point in direction -90 move 60 steps point in direction 180 move 120 steps point in direction -90 move 30 steps					

In terms of coding, typically, problems could require learners to

- read code and explain what it does
- work through (trace) / act out code (physically or simulated) to determine the output or the correctness
- complete code instructions (code instructions are provided with some instructions or code elements missing) that learners need to complete
- translate verbal/written instructions (algorithm) to code (e.g. write block-based code for a list of symbolic (e.g. arrows)/written instructions))
- add functionality/instructions to an existing program
- rewrite a set of coding instructions to be more efficient, e.g. using a loop construct for code that is repeated
- choose the correct solution from 2-3 options
- compare different solutions to evaluate efficiency
- debug an algorithm or block-based program (find the bug, describe the bug, and correct it)
- develop a solution/algorithm (code instructions) based on a given problem or for an open-ended problem through planning, implementing, testing, and debugging.

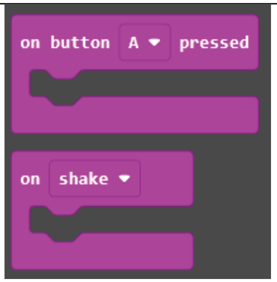
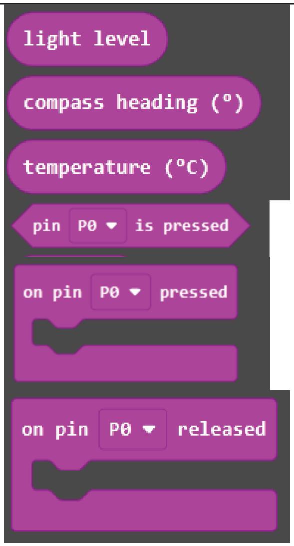
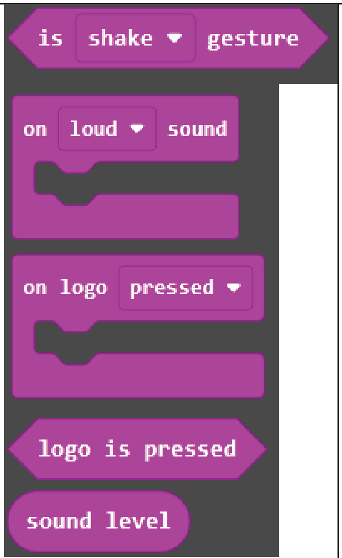
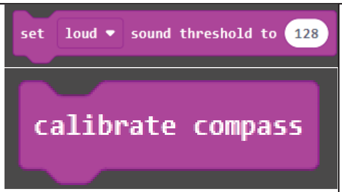
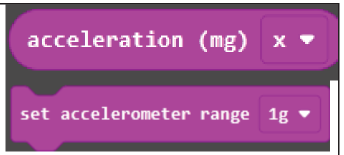
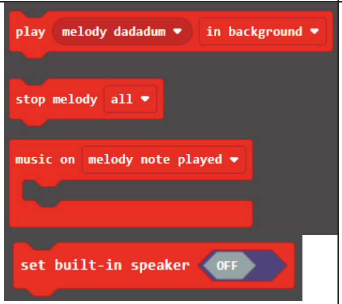
depending on the competency/(ies) the learner needs to demonstrate.

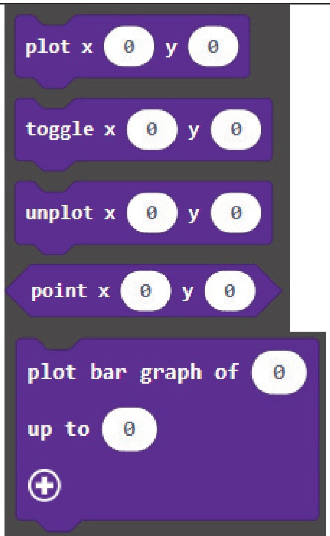
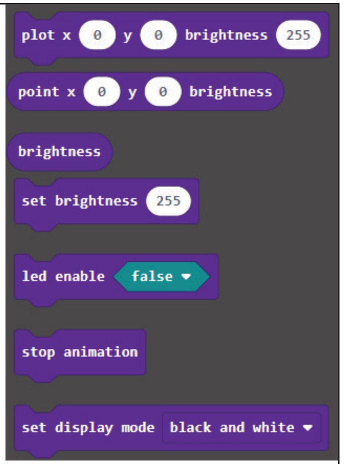
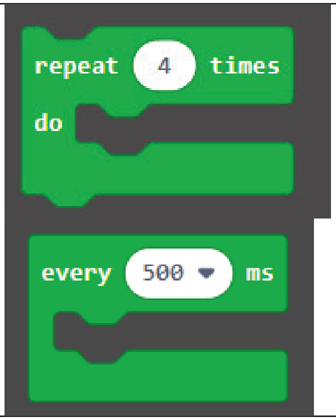
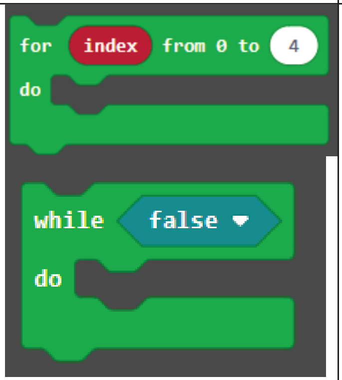
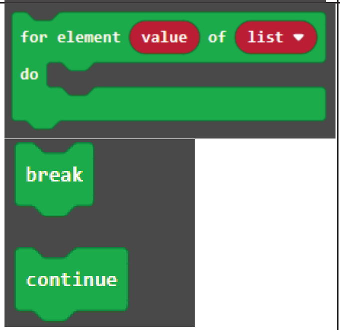
2.16.2 Robotics

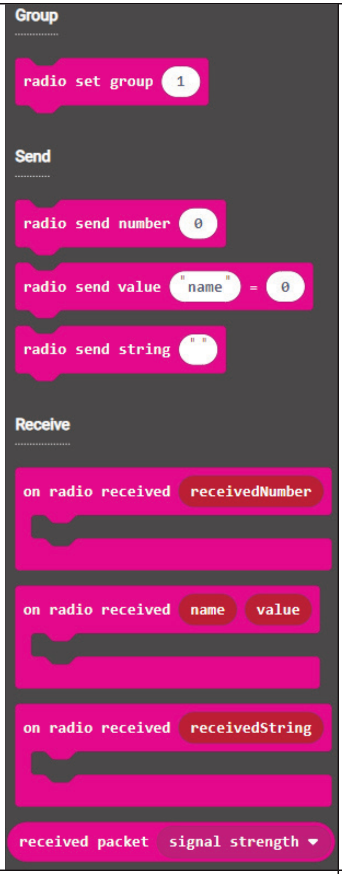
The following table provides the robotics competencies that learners must demonstrate by the end of each Grade from Intermediate to Senior Phase:

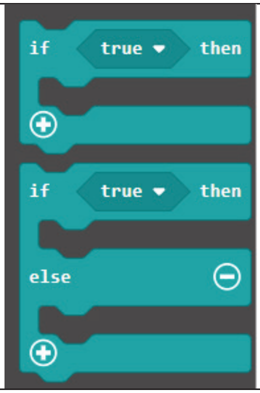
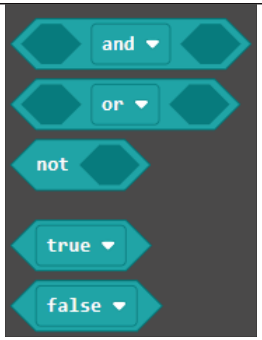
Table 2-8 Senior phase robotics coding concepts, content and skills breakdown and progression per grade

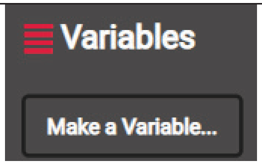
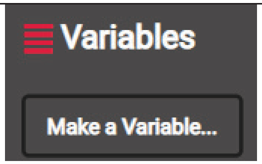
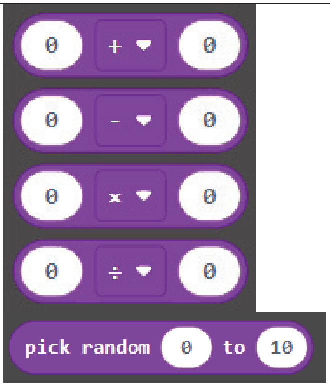
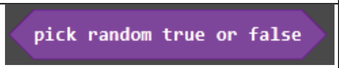
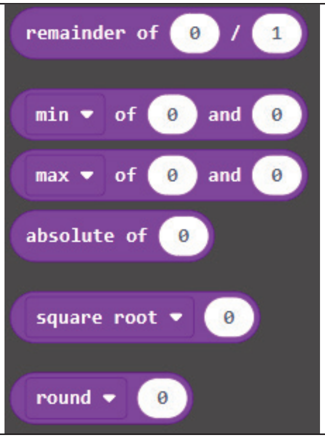
	MakeCode (Micro: bit) for Intermediate and Senior phase (content breakdown and concept progression)					
	Grade 4	Grade 5	Grade 6	Grade 7	Grade 8	Grade 9
▦ Basic						

Input		 on button A pressed on shake Shake = sensors describe  button A is pressed Clearly differentiate between the on (event) and the if as a conditional construct.	 light level compass heading (°) temperature (°C) pin P0 is pressed on pin P0 pressed on pin P0 released	 is shake gesture on loud sound on logo pressed logo is pressed sound level	 set loud sound threshold to 128 calibrate compass	 acceleration (mg) x set accelerometer range 1g
Music		 Melody play melody at tempo Tone play tone Middle C for 1 beat rest(ms) 1 beat set volume 127 volume stop all sounds	 change tempo by (bpm) 20 set tempo to (bpm) 120 1 beat tempo (bpm)	 play melody dadadum in background stop melody all music on melody note played set built-in speaker OFF		

Led					 plot x 0 y 0 toggle x 0 y 0 unplot x 0 y 0 point x 0 y 0 plot bar graph of 0 up to 0	 plot x 0 y 0 brightness 255 point x 0 y 0 brightness brightness set brightness 255 led enable false stop animation set display mode black and white
Loops		 repeat 4 times do every 500 ms	 for index from 0 to 4 do while false do Check while		 for element value of list do break continue	

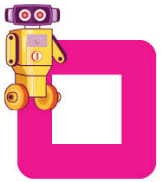
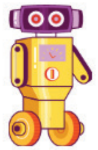
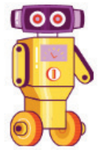
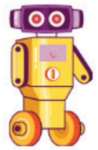
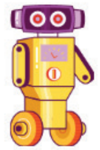
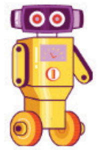
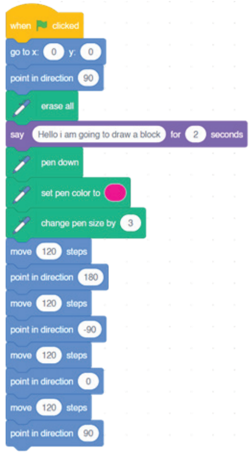
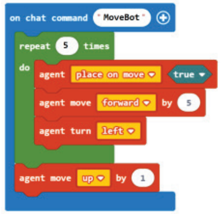
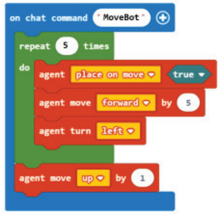
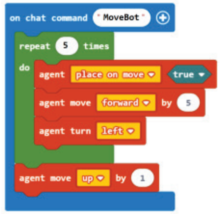
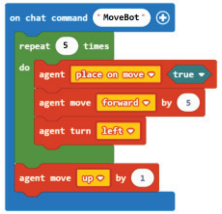
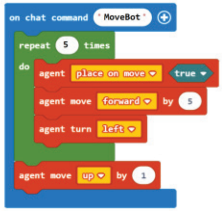
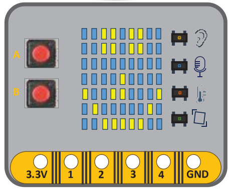
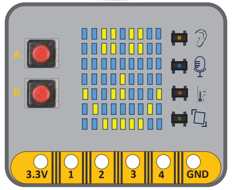
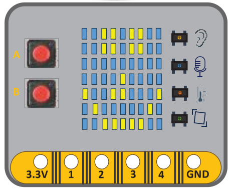
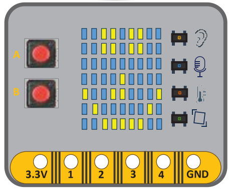
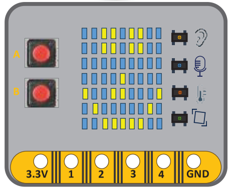
Radio				Group radio set group 1 Send radio send number 0 radio send value "name" = 0 radio send string " " Receive on radio received receivedNumber on radio received name value on radio received receivedString received packet signal strength		

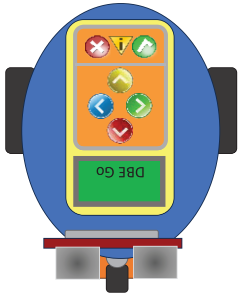
Logic		if true then
		if true then
		else
			and
			or
			not
			true
			false			

		 0 = 0 0 < 0				
Variables			 Variables Make a Variable...	 Variables Make a Variable... Multiple variables may be introduced with some guidance in the problem statement		
Math		 0 + 0 0 - 0 0 x 0 0 ÷ 0 pick random 0 to 10	 remainder of 0 ÷ 1 round 0	 pick random true or false	 remainder of 0 / 1 min of 0 and 0 max of 0 and 0 absolute of 0 square root 0 round 0	 constrain 0 between 0 and 0 map 0 from low 0 high 1023 to low 0 high 4
Extenstions		 microturtle	 servo			
Application Skills (IDE)		Save your program Open an existing program (Change a given application)	Download a program to the device (Hex file)			

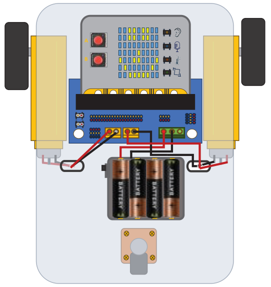
2.16.3 Robotics progression Grade 4 – Grade 9

Table 2-9 Progression of Robotics and components per grade

Grade 4	Grade 5	Grade 6	Grade 7	Grade 8	Grade 9
					
Virtual robot in a code/simulated environment	***Virtual robot*** in a code/simulated environment	***Virtual robot*** in a code/simulated environment	***Virtual robot*** in a code/simulated environment	***Virtual robot*** in a code/simulated environment	***Virtual robot*** in a code/simulated environment
 when 🏁 clicked go to x: 0 y: 0 point in direction 90 erase all say Hello i am going to draw a block for 2 seconds pen down set pen color to change pen size by 3 move 120 steps point in direction 180 move 120 steps point in direction -90 move 120 steps point in direction 0 move 120 steps point in direction 90	 on chat command "MoveBot" repeat 5 times do agent place on move true agent move forward by 5 agent turn left agent move up by 1	 on chat command "MoveBot" repeat 5 times do agent place on move true agent move forward by 5 agent turn left agent move up by 1	 on chat command "MoveBot" repeat 5 times do agent place on move true agent move forward by 5 agent turn left agent move up by 1	 on chat command "MoveBot" repeat 5 times do agent place on move true agent move forward by 5 agent turn left agent move up by 1	 on chat command "MoveBot" repeat 5 times do agent place on move true agent move forward by 5 agent turn left agent move up by 1
Block based coding for robot control	Block based coding for robot control	Block based coding for robot control	Block based coding for robot control	Block based coding for robot control	Block based coding for robot control
Optional Educational Robot –					
	Microcontroller Interact with onboard sensors and onboard output components only.	***Microcontroller*** Interact with onboard sensors and onboard output components. **ONE additional hardware** component e.g. 	***Microcontroller*** (or TWO Microcontrollers for – **BASIC** Radio Communication) [Simple send and receive App]	ONE ***Microcontroller*** (or TWO Microcontrollers for - Radio Communication)	ONE ***Microcontroller*** (or TWO Microcontrollers for - Radio Communication) Interact with onboard sensors and onboard output components.



(for enrichment with block-based coding)

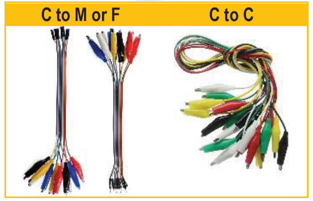


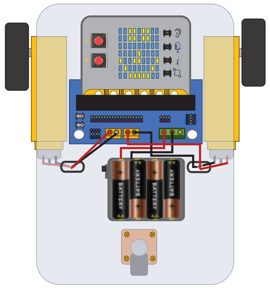
Educational robot (for enrichment with block-based coding)

Resistor

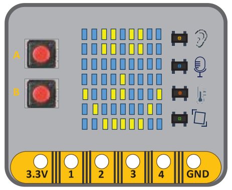
Push-Button

Connections with crocodile clips **only** (NO soldering and no breadboards)



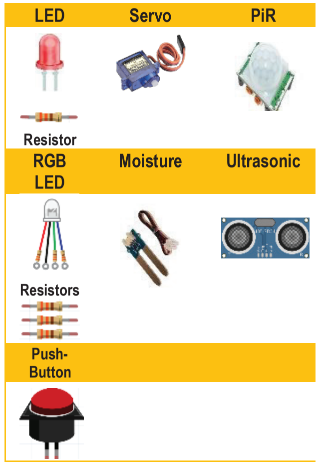


Educational robot (for enrichment with block-based coding)

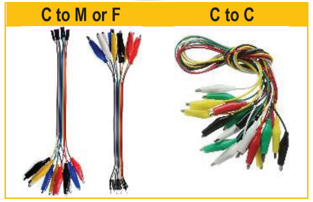


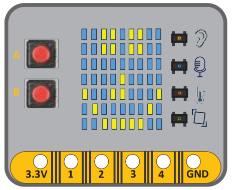
Interact with onboard sensors and onboard output components.

ONE (**or TWO** additional hardware components (Maximum)) e.g.

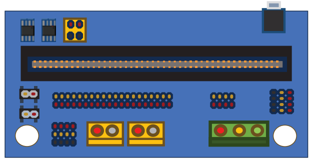


Connections with crocodile clips **only** (NO soldering and no breadboards)



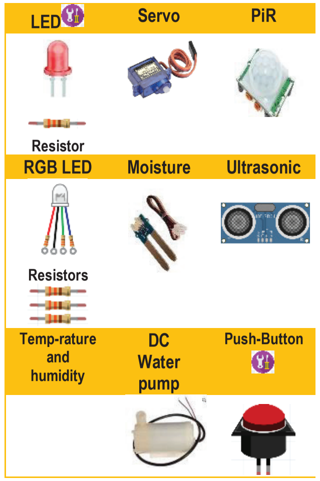


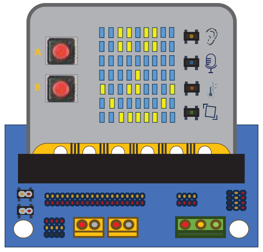
Interact with onboard sensors and onboard output components.



ONE, TWO OR TRHEE additional hardware components (Maximum **THREE**) Connected to Microcontroller – Or breakout board) e.g.

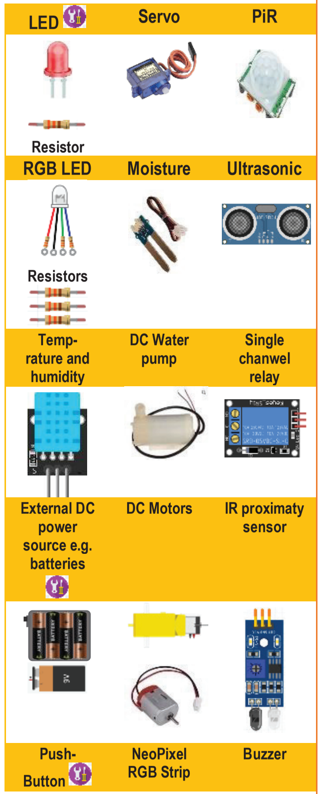
The following icon indicates that the component is covered in the technology curriculum. Electronic components should be described and implemented in an elementary level.

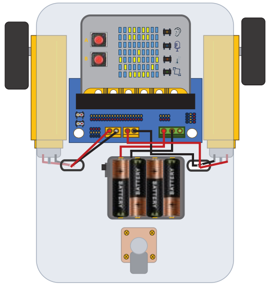




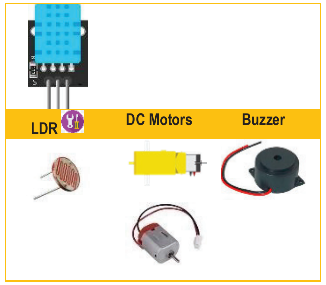
Breakout board & motor driver

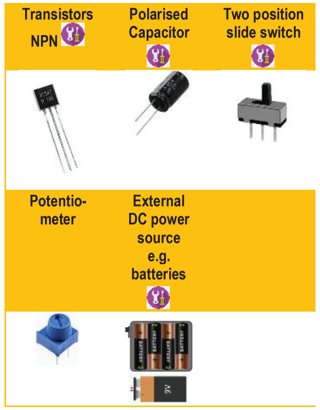
ONE, to FOUR additional hardware components (Maximum **FOUR**) Connected to Microcontroller – Or breakout board) e.g.





Educational robot

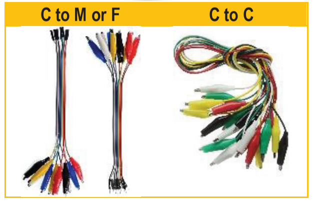


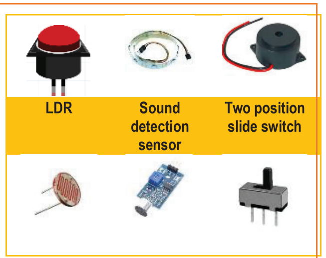


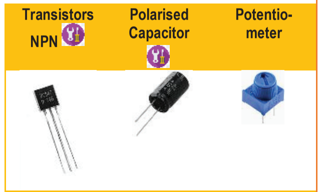
IMPORTANT

Electronic components should be described and implemented in an elementary level.

Connections with crocodile clips and/or simple breadboard circuits



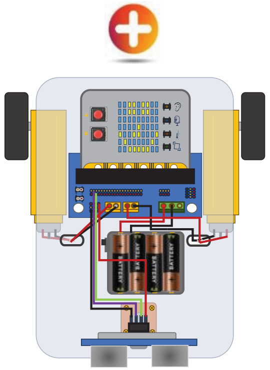
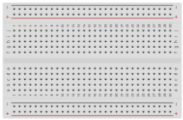
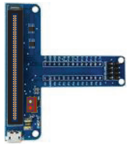
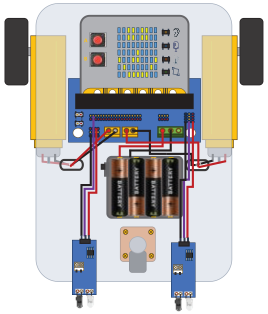
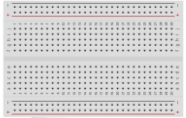
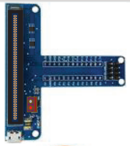


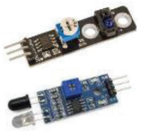


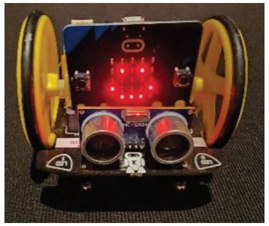
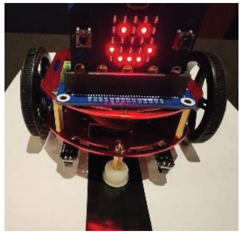
IMPORTANT

Electronic components should be described and implemented in an elementary level.

Connections with crocodile clips and M2F and M2M and/or Jumper wires and/or breadboard

				 Educational robot + Breakout board & motor driver (For Obstacle avoidance using Ultrasonic sensor only) Simple breadboard circuits  	 Educational robot + Breakout board & motor driver + Line following robot with 2 IR proximity sensors Simple breadboard circuits   **Optional components – (For enrichment)** **7 Segment Display**  **8x8 Led Display**  **LCD 16x2 Display**   **Resistors**
	Simple Sample Projects Based on using board only	**Simple Sample Projects** **(Single component only + Onboard components)** ▪ Soil Moisture with two conductors (e.g., nails)	**Simple Sample Projects** **(One or two components + Onboard components)** ▪ Soil Moisture with nails and (sensor) ▪ RGB LED Project	**Sample Projects** **(One or two components + Onboard components) + Breadboard (if required)**	**Sample Projects** **(Four components Max + Onboard components) + Breadboard (if required)**

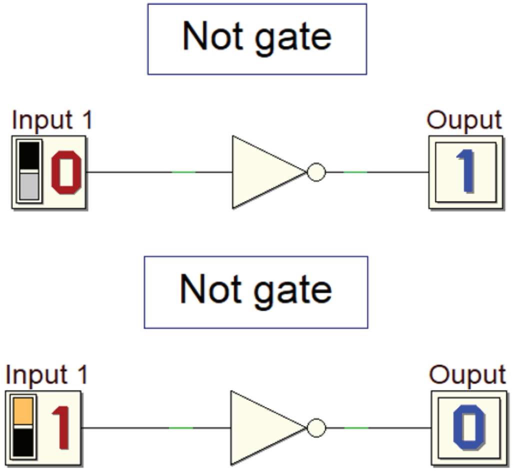
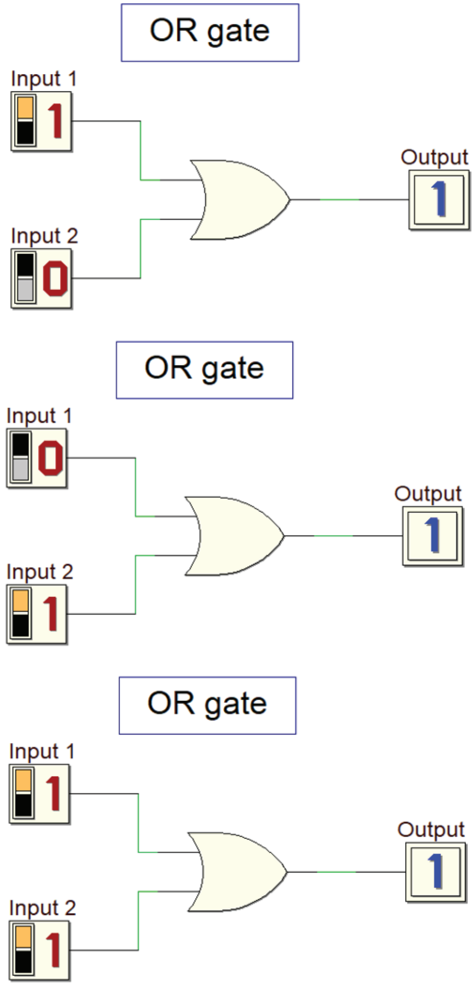
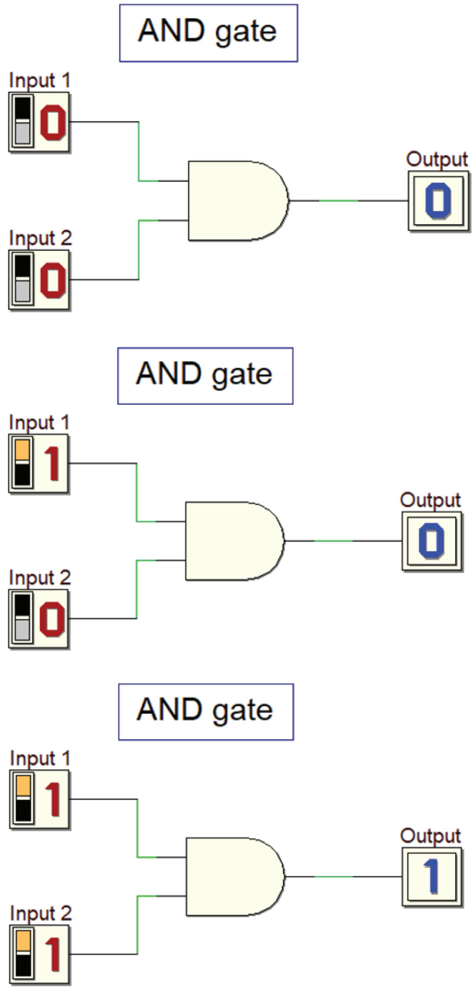
	- Monster Munch (Servo project) - Own switch with foil (LED Project) - Simple alarm with PIR and onboard buzzer - Traffic light with 3LED's	- Ultrasonic sensor Project - Alarm with PIR or Ultrasonic sensor and additional component e.g., External buzzer, LED.	- "Smart device" (Combination of sensor and responder device) - Automatic dustbin or opener (railway crossing sensor and servo) - Motor boom gate + Breakout board. - Obstacle avoidance robot	- Soil moisture with simple DC water pump + Relay + (Self-watering plant) - Smart home / Greenhouse (2 Sensors + "2 Actions". E.g., PIR and auto light on & Temperature and auto fan on and Relay where required) - "Smart device" (Combination of sensors and responder device and output display) - Line following robot
			Obstacle avoidance robot (OAR) using an ultrasonic sensor.  The inclusion of the OAR is to introduce the learners to the practical principles behind the operation of the robot. It is not expected of the learners to construct the robot, only to describe its basic principles of operation and the code logic behind it. **The basic principles behind the operation of an obstacle avoidance robot using an ultrasonic sensor.** An ultrasonic sensor emits a sound wave and measures the time it takes for the wave to return, which is used to calculate the distance to an obstacle. The robot uses this information to avoid obstacles by changing its direction of travel when too close. **Any applicable kit could be used to illustrate the operation of the robot.**	**Line following robot (LFR) using IR sensors**  The inclusion of the LFR is to introduce the learners to the practical principles behind the operation of the robot. It is not expected of the learners to construct the robot, only to describe its basic principles of operation and the code logic behind it. **The basic principles behind the operation of a line following robot using IR sensors.** IR sensors detect the presence of a line by measuring the amount of infrared light reflected from the surface. The robot uses this information to follow the line by adjusting the speed and direction of its motors. In more detail, the IR sensors emit infrared light and measure the amount of light that is reflected. When the sensor is over a black line, the light is mostly absorbed and very little is reflected. When the sensor is over a white line, more light is reflected.

					The robot uses this information to follow the line by adjusting the speed and direction of its motors **Any applicable kit could be used to illustrate the operation of the robot.** 
		Elementary circuits and projects →			
			Circuits and elementary breadboard projects →		

2.16.4 Truth Tables and Logical Gates

In Grade 8 and 9, learners are exposed to the following logical gates:

Table 2-10 Logic gates Grade 8 and Grade 9

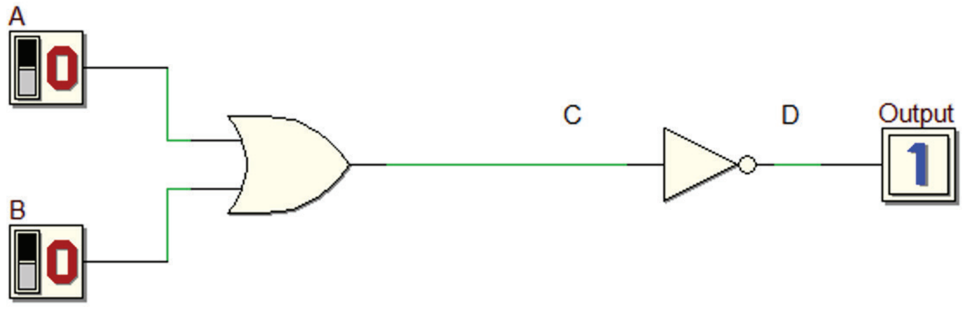
NOT gate	OR gate	AND gate
		

Learners will be exposed to the following truth tables:

Table 2-11 Truth Tables Grade 8 and Grade 9

Grade 8

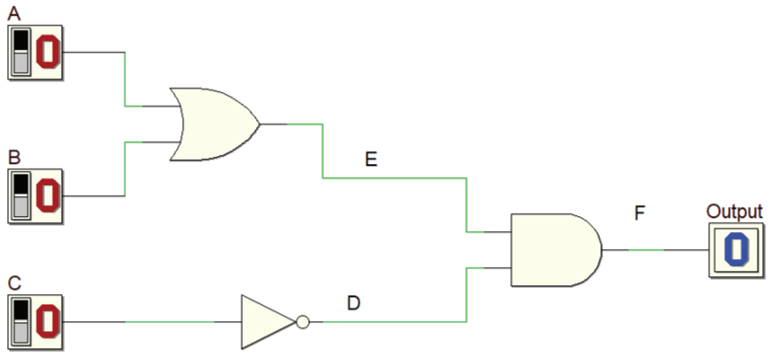
Two inputs only combined with a maximum of three gates



A	B	C = A OR B	D = NOT C	D = Output
0	0	0	1	1
0	1	1	0	0
1	0	1	0	0
1	1	1	0	0

Grade 9

Three inputs on/y combined with a circuit with a maximum of four gates



A	B	C	D = NOT C	E = A OR B	F = D AND E	F = Output
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	1	0	1	1	1	1
0	1	1	0	1	0	0
1	0	0	1	1	1	1
1	0	1	0	1	0	0
1	1	0	1	1	1	1
1	1	1	0	1	0	0

3 SECTION 3
CONTENT SPECIFIC CLARIFICATION PER GRADE PER TERM

The following tables provide the content clarification per term and per grade.

This section should be read in conjunction with Tables 2-1 to 2-11 and Figures 2-7 to 2-11

In Senior Phase, the curriculum is designed to also strengthen the specific concepts and content that link to other subjects such as Mathematics, Natural Sciences, Technology, Design and Life Skills.

Content clarification is done with examples as Coding and Robotics is a new subject.

Note:

This section contains examples that clarify the content and competencies. These examples serve as illustrations to better understand the topics and the competencies learners are expected to develop.

However, teachers should see these examples as a starting point for teaching the content and competencies. While the examples are beneficial, teachers should not limit themselves to just those activities. They are encouraged to include other exercises and tasks to ensure deliberate practise, retrieval practice and a deeper understanding of the concepts and skills being taught.

The content and competencies are also grouped based on the main topic areas. This organisation helps teachers understand which skills and knowledge are related and how they are connected. The content and competencies are therefore not necessarily listed in the order they must be taught. Teachers have flexibility in how they sequence the topics based on the context of their teaching environment and the needs of their leaners. However, there is an indication of how different competencies relate to each other. This linkage could help teachers understand the progression of skills and how they support or build upon one another or could be taught in relation with other skills and competencies.

Teachers should therefore develop their Annual Teaching Plans (ATPs) sequencing content and competencies in a manner that will make sense for their learners and their teaching and learning environment to foster a positive learning experience. The goal of developing the ATPs is to maximize the learners' learning outcomes, acquisition of competencies and achievement.

It is also important to note that physical and paper-based activities should not be neglected once learners start to work on a computer,

3.1 GRADE 7

Note:
Teachers must include the following competencies and content in their Annual Teaching Plan (ATP), distributed across the terms and sequenced, organised and grouped in a manner that will facilitate learning in a manner that will make sense for learning and teaching, maximize the learners' learning outcomes and achievement. and in a way that will make optimal use of time and resources. Certain competencies could also be combined in bigger/more complex activities.

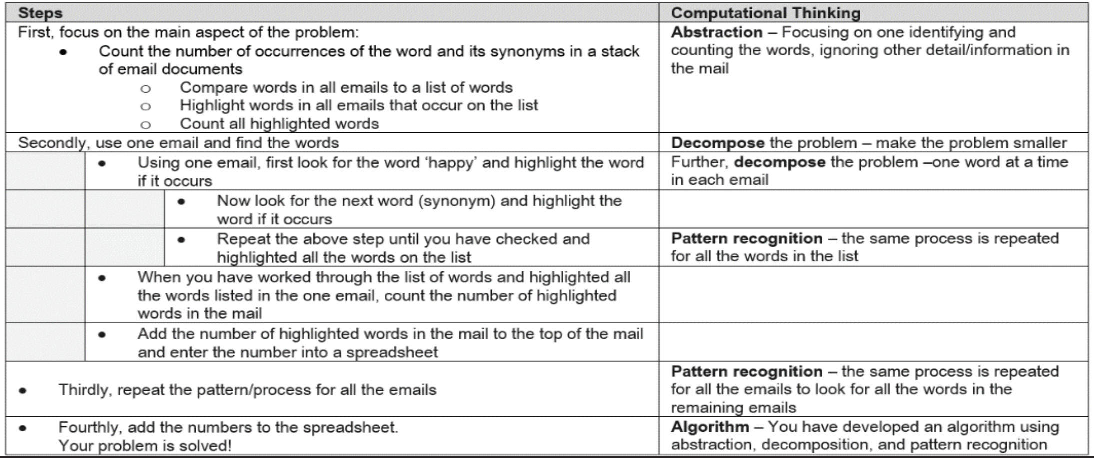
3.1.1 Term 1

<table>
<tr><th>Content (Grade 7 / Term 1)</th><th>Notes/Examples</th></tr>
<tr><td>Coding</td><td></td></tr>
<tr><td>C.1 Apply computational thinking (CT) skills to develop a set of logical instructions to solve a problem.</td><td>Link to C.2</td></tr>
<tr><td>

Example Activity 1 – Solve a problem in pairs
The stationery shop at school is marketing a specific product and encourages learners to write a review on the product and send their review via email. The school has received many emails containing reviews on the product and the principal wants to analyse these to see how many learners seem to be happy with the product and whether they need help. To be able to do so, you need to determine how many times the word 'happy' or any synonym/word that indicates learner satisfaction occurs in all the mails.
The synonyms that you will be looking for are excellent, satisfied, good, pleased, delighted, superb, great, and nice.
To be able to find a solution, the following questions will help:

- What are the main aspects to focus on?
- What are the main sub-problems that you need to solve in solving the overall problem?
- What patterns can you see in the solution, i.e., what steps/processes need to be repeated? Can the patterns be generalised?
- What is the algorithm that you would use to solve this problem?

You should only focus on the general problem of counting the number of occurrences of the word and its synonyms in a stack of printed email documents.
In pairs, work through the following steps (the driver simulating/acting out the steps and the navigator instructing the driver on which steps to follow and checking if the correct steps are followed) and test them to see if they will solve the problem. If not, change them to solve the problem. The table below will provide a guide on the steps you can follow to solve the given problem:



<table>
<tr><th>Steps</th><th>Computational Thinking</th></tr>
<tr><td>First, focus on the main aspect of the problem:
• Count the number of occurrences of the word and its synonyms in a stack of email documents
○ Compare words in all emails to a list of words
○ Highlight words in all emails that occur on the list
○ Count all highlighted words</td><td>Abstraction – Focusing on one identifying and counting the words, ignoring other detail/information in the mail</td></tr>
<tr><td>Secondly, use one email and find the words</td><td>Decompose the problem – make the problem smaller</td></tr>
<tr><td>• Using one email, first look for the word 'happy' and highlight the word if it occurs</td><td>Further, decompose the problem –one word at a time in each email</td></tr>
<tr><td>• Now look for the next word (synonym) and highlight the word if it occurs</td><td></td></tr>
<tr><td>• Repeat the above step until you have checked and highlighted all the words on the list</td><td>Pattern recognition – the same process is repeated for all the words in the list</td></tr>
<tr><td>• When you have worked through the list of words and highlighted all the words listed in the one email, count the number of highlighted words in the mail</td><td></td></tr>
<tr><td>• Add the number of highlighted words in the mail to the top of the mail and enter the number into a spreadsheet</td><td></td></tr>
<tr><td>• Thirdly, repeat the pattern/process for all the emails</td><td>Pattern recognition – the same process is repeated for all the emails to look for all the words in the remaining emails</td></tr>
<tr><td>• Fourthly, add the numbers to the spreadsheet.
Your problem is solved!</td><td>Algorithm – You have developed an algorithm using abstraction, decomposition, and pattern recognition</td></tr>
</table>

</td><td>

Remind learner about the guidelines for developing an algorithm

- Understand **the problem**
  - Read the problem statement carefully to understand what the problem is that you need to solve
- Analyse **the problem**
  - What should the output (result) be?
  - What are the inputs needed to produce the output?
  - What processing (if any) needs to be done to get to the output?
  - Can the problem be broken down into smaller problems? (Decomposition)
  - Are there any patterns? (Pattern recognition)
- Develop **a high-level algorithm**
  - Determine the main steps, ignoring the detail
- Refine the algorithm by decomposing steps (breaking steps down into smaller tasks) or instructions and adding more detail
- **Test** the algorithm – follow the instructions to see if it delivers the desired output
  Does the solution or output make sense?
  Is the sequence logical?
  Is the algorithm efficient?
  Are all parts of the problem or task covered?
  Are some instructions or tasks repeated unnecessarily?
- Evaluate the algorithm and **update** or correct (debug) the algorithm if necessary.
- Translate the algorithm into code.

</td></tr>
</table>

Content (Grade 7 / Term 1) | **Notes/Examples**

C.2 Present a simple coding solution using symbolic or written statements representing sequences of commands, single repetition, and conditional constructs.

Link to D.10 and D.1
(Start with D10 before doing C.2)

Example activity 1 – Division and remainder calculator

Sfiso bought a box of apples and must divide the apples between the learners in a class.

Write a block-based program that will help him to divide the box of apples fairly between his classmates by doing the following:

Determine how many apples each learner in the class will receive if a box of apples is divided between the learners in the class and how many apples will remain (enforce the concepts of division and remainder (mod).

- The program must input the number of apples in the box and the number of learners in a class.
- The program must display how many apples each learner will get and the number of apples that will be left in the box.

Revise. numeric operators and calculations done for example activity.

Revise input, processing and output in coding

Revise coding concepts from previous grades to match activities given.

Learners need to apply

- numeric operators in writing programs with various calculations
- relational and Boolean operators to enable comparison
- basic string (character) operator, e.g. join to provide output

C.3 Interpret and execute a given symbolic or written set of commands

Link to C.1, C.2 and R.6 and D.8 -D.8 and D.9

Example activity – Multiplication Game

(Revise IF with FOREVER loop and calculations)

Study the code on the right

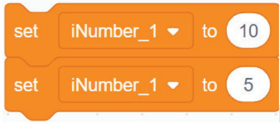
Explain what the program does

Run the program and see if you predicted correctly

Note:

Ensure learner understand variables correctly as many learners tend to struggle with the concept of a variable and tend to form misconceptions about variables. If learners gain misconceptions about variables, it is likely that they will also struggle to master other programming concepts like branching or repeating (Zanko et al, 2019). Some misconceptions include, e.g.:

- **Assignment** (*set…to (answer or other value)*), e.g.

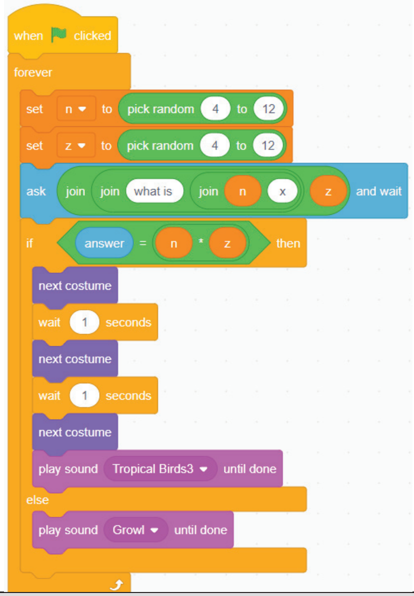


```
set iNumber_1 ▾ to 10
set iNumber_1 ▾ to 5
```

Learners may belief that iNumber will contain the first assigned value (10)

Or maybe the sum of the two values (15)

- **Output** (e.g., *say…): *Learners may believe that the variable name rather than its value will be outputted (say…)



```
when 🏴 clicked
forever
    set n ▾ to pick random 4 to 12
    set z ▾ to pick random 4 to 12
    ask join join what is join n x z and wait
    if answer = n * z then
        next costume
        wait 1 seconds
        next costume
        wait 1 seconds
        next costume
        play sound Tropical Birds3 ▾ until done
    else
        play sound Growl ▾ until done
```

Revise variables

Revise control structures done in previous grades

Learners need to apply

- Sequential statements (execution of code line by line, from top to bottom, in the order it appears)
- Conditional statements (branching) and conditions (allow the program to make decisions based on specific conditions (true or false))
- Loop statements (iteration) (to repeatedly execute a block of code based on a constant value or a simple condition).

C.4 Debug a given symbolic or written set of instructions.

Example activity – Find and correct the bug

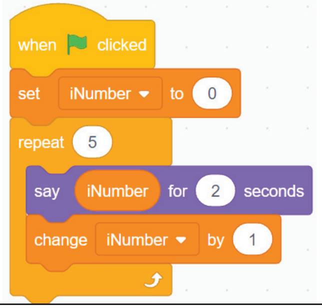
The program on the right must display the

Simple trace table for the problem

first five natural numbers (1 – 5).

However, it does not work correctly.

Work through the program (use a trace table) to find and correct the error.



```
when 🏴 clicked
set iNumber ▾ to 0
repeat 5
    say iNumber for 2 seconds
    change iNumber ▾ by 1
```

Step	iNumber	Loop Counter	Output (say) iNumber	Test (Loop executed 10x)?
1	0	1	0	
	1	1		No
2	1	2	1	
	2	2		No
3	2	3	2	
	3	3		No
4	3	4	3	
	4	4		No
5	4	5	4	
	5	5		Yes → Stop

Note:

There is more than one way to correct the code, e.g.

set iNumber to 1 or

swop the two statements within the loop (first change iNumber then say iNumber

Discuss both ways of correcting with learners. It is important that learners understand why both corrections can work.

Show learners how to do a simple trace table to trace code to determine the output.

<table>
<tr><th>Content (Grade 7 / Term 1)</th><th>Notes/Examples</th></tr>
<tr><td>C.5 Evaluate a given solution towards potential improvement.</td><td></td></tr>
<tr><td>Example activity
Discuss the two ways of correcting the program in C.4 (display the first 5 natural numbers).
Evaluate both ways of correcting. Is one way of correcting better than the other?</td><td>Evaluate different ways of achieving an outcome.</td></tr>
<tr><td>C.6 Recognise and interpret patterns in symbolic sets of data or visualisations.</td><td>Link to C.1, C.2, C.3 and D.8, D.9</td></tr>
<tr><td>Example activity
You have done the algorithm for a square (4 corners, 4 sides) and a triangle (3 corners, 3 sides) in previous grades.
Now look at other geometrical figures: pentagon, hexagon, heptagon and octagon. Could there be a pattern that indicates a relationship between all these figures?
Complete the following table to determine a possible relationship
<table>
<tr><th>Figure</th><th>Square</th><th>Pentagon</th><th>Hexagon</th><th>Heptagon</th><th>Octagon</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Number of sides</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr>
<tr><td>Number of angles</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr>
<tr><td>Size of interior angle</td><td>90°</td><td>72°</td><td>60°</td><td>51,4°</td><td>45°</td></tr>
<tr><td>Sum of interior angles</td><td>360°</td><td>360°</td><td>360°</td><td>360°</td><td>360°</td></tr>
</table>
Answer the following questions:
1. What do you notice about the form of the shapes when the number of sides increases?
2. What happens with the size of the interior angles when the number of sides of the figures increases?
3. What do you notice about the sum of the interior angles?
4. Describe and explain the pattern/relationship.</td><td>Pattern recognition is based on the five key steps:
• Identifying common elements in problems or tasks
• Identifying and interpreting common differences in problems or tasks
• Identifying individual elements within problems or tasks
• Describing patterns that have been identified

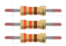
Pattern – The way in which something is arranged (usually with repetition) or the way in which it happens or is done.
Predictions are the connecting links between what you already know, with new information or knowledge
Pattern recognition is therefore the process of recognising patterns, e.g. in data or behaviour.
One makes predictions based on identified patterns.</td></tr>
<tr><td>Robotics</td><td></td></tr>
<tr><td>R.1 Explain what a robot is in simple terms.</td><td>Link to R.2 and R.3</td></tr>
<tr><td>Example activity – What is a robot?
Divide learners into pairs and provide each pair with printed material of videos on robots as well as a worksheet with questions (see notes column) they need to discuss and answer,
Each pair present their answer to the class when done.
Teacher consolidates answers,</td><td>Learners must be able to
• define a robot (what it is)
• describe the purpose of a robot
• describe the contexts they operate in.
• understand concepts regarding the relationship between the composition of a robot - basic parts (sensors, controllers, actuators, power source)
• understand the evolution of robots / advancements of robots (also link to the concept of AI –elementary reference to automatic decisions)</td></tr>
<tr><td>R.2 Identify different types of robots.</td><td>Link to R.1</td></tr>
<tr><td>Discuss the types of robots with learners (simple ta-oriented robots, chatbot (in relations to automation and virtual robots used in the information system space))
Example activity – Investigate a robot used in the hospitality industry
Divide learners into pairs. Learners:
• investigate a robot used in the hospitality industry
• outline the main components of the selected robot.
• describe how the robot is used to achieve its goals.
• create a diagrammatical outline of the robot, and its various components.
• create a concise presentation regarding how robots are used in the hospitality industry that could also be used in a hospital (medical environment).</td><td>R.1 and R.1 can be done together
Learners must outline different types of robots in terms of their use and application, i.e.
• Simple Task-Oriented Robots
• Chatbots (in relation to automation and virtual robots used in the information systems space)</td></tr>
</table>

Content (Grade 7 / Term 1)

R.3 Outline the different components of a robot

Example activity – Tell and show

Discuss the following components in terms of basic definition, purpose and practical application:

Component	Basic definition	Purpose	Practical application example – May be used to:
Push-Button 	A simple mechanical switch that is pressed to complete an electrical circuit.	Used as an input device to trigger specific actions or functions in a robot.	Functions as a user interface element for manual control or emergency stop in a robot.
Resistor: 	A passive electrical component that limits the flow of electric current.	Utilized to control current, voltage, and protect other components from excessive current in a circuit.	Adjusts current levels in various parts of the robot's circuitry, maintaining proper operation.
Two-Position Slide Switch 	A switch with two positions (on/off) that can be physically slid between the two states.	Provides a manual way to control power or functionality in a robot by toggling between two settings.	Employed to toggle power sources, modes, or operation states in a robot.
Buzzer 	An electronic sound-producing component.	Used to generate audible alerts, notifications, or feedback in a robot.	Provides auditory feedback to users or alerts in response to specific events or conditions.
LED (Light-Emitting Diode) 	A semiconductor device that emits light when current flows through it.	Used for visual indicators, status lights, and displays in robots.	Indicates the status of various robot systems, such as power, connectivity, or task progress.
DC Motors  	Electric motors that operate on direct current (DC) power.	Provide motion to various parts of a robot, such as wheels, arms, or grippers.	Powers robot movement, including wheel propulsion, arm articulation, and conveyor systems.
External DC Power Source (e.g. batteries) 	An independent source of electrical power, typically in the form of batteries.	Provides the primary source of energy to drive all robot components and systems.	Provides the necessary energy to operate the robot's components and systems.

R.5 Design a simple product (artefact) based on a set of design specifications.

Example activity (project) – Steady hand game (No breadboard)

This game involves the use of simple electronic components (Buzzer, Battery, Wire, and Crocodile clips)

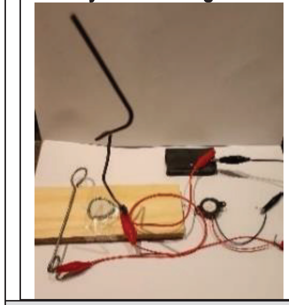
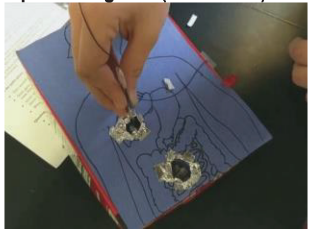
Divide the learners into pairs.

The learners design their own maze and connect the buzzer and the battery. (The Maze wire may be bend in different ways).

Code is written on a micro: bit (which is not connected to the board) to keep score.

When Button A and B is pressed the game starts and all scores reset. The player then moves the wire, and the second player keeps score using the micro: bit by pressing the A button if the buzzer sounds. When Button A is pressed the micro: bit should also buzz indicating that the penalty was captured.

When Player A is done, it is Player B's turn. After the game, the Logo is pressed on the micro: bit, and the winner is indicated.

Steady hand wire game - Example 1	Steady hand wire game - Example 2	Operation game (Derivative)
	 See: https://www.instructables.com/Buzz-Wire-Kit/	 https://learn.adafruit.com/low-tech-operation-game/overview

R.7 Create, test, and execute a set of robotic instructions.

Notes/Examples

Link to R.1, R.2 and R.5

Review the following electronic components in terms

- Basic definition
- Purpose, and
- how it could be used in simple physical computing and robotics projects.

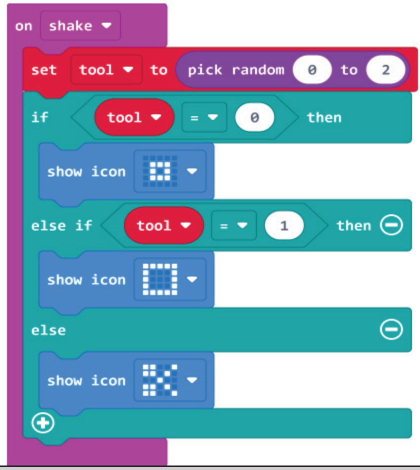
IMPORTANT in GRADE 7 all robotics / physical computing projects are con onboard structed to interact with sensors and onboard output components of the microcontroller plus ONE (or TWO additional hardware components that may include sensors and actuators (Maximum)). All connections with crocodile clips (or derivatives) only (NO soldering and no breadboards) Learners are guided on the use and configuration of the components (including the sensors and actuators)

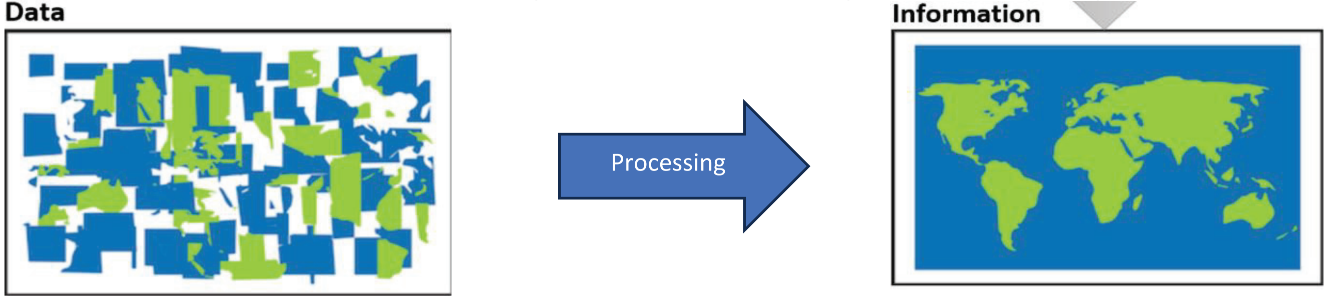
Note:

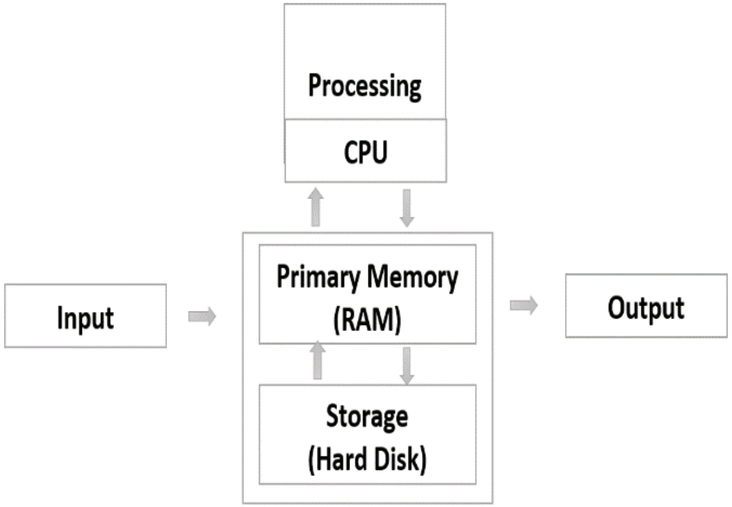
In grade 7 the learners should know the basic purpose of a resistor. The correct resistor/s for a project should be supplied. The learners need not know how to interpret the colour codes.

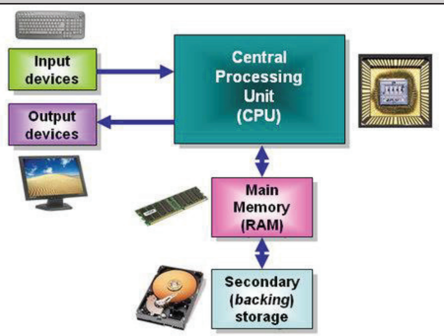
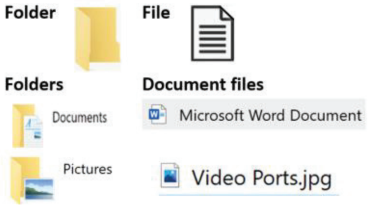
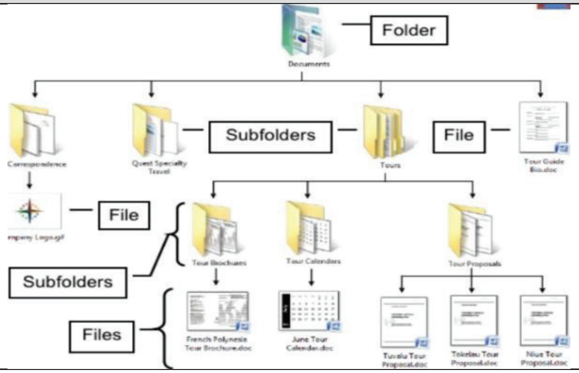
Link to R.3

Content (Grade 7 / Term 1)	Notes/Examples

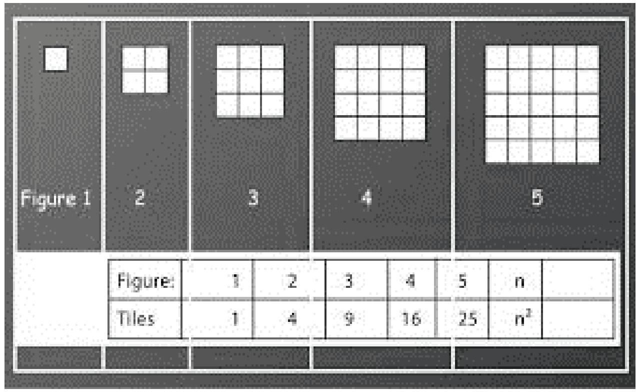
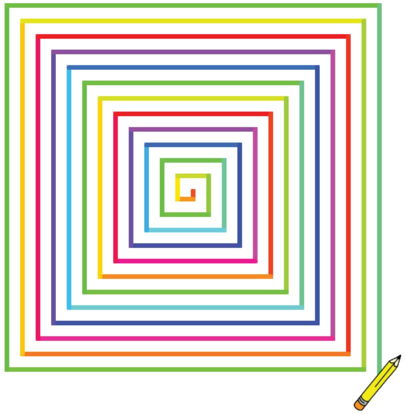
Example activity 1 - Rock-papers-scissor game artefact (level 1)
In pairs, design a rock-paper-scissors game artefact
When the microcontroller is shaken, it must randomly display a "rock", "paper" or "scissors" icon.
Tip: Select a random number between 0 and 2. Based on the random number, display rock, paper or scissors. Pairs play against each other. One learner in each pair keeps score. The pair who first gets three points, wins.
https://microbit.org/projects/make-it-code-it/rock-paper-scissors/
Example activity 2 - Rock-papers-scissor game artefact (level 2) (extend from activity 1)
Add two variables to keep count of each player's points and to display a message when a team has won. When Team A scores, button A will be pressed to increase the score. When Team B scores, button B will be pressed to increase their score.
Tips for solution:
• When Button A is pressed, Team A's points must increase.
• When Button B is pressed, Team B's points must increase.
If a team has reached 3 points, make a "beep" sound and display a message to say which team has won.

`on shake`
`set tool to pick random 0 to 2`
`if tool = 0 then`
`show icon`
`else if tool = 1 then`
`show icon`
`else`
`show icon`
	You want to interleave practise in problems.
	"It is important that problem types must differ, for example, you want to randomly have a problem of one type and then solve a problem of another type and then a problem of another type. And in doing that, it feels difficult, and it doesn't feel fluent. And the signals to your brain are, I am not getting this. I am not doing very well. But in fact, that effort to try to figure out what kinds of approaches do I need for each problem as I encounter a different kind of problem, that is producing learning. That is producing robust skills that stick with you."
	Dr Mark A. McDaniel, Harvard University

Digital Concepts	
D.1 Outline the concept of technology and purpose of information technology (IT)	Link to D.3 and D.7
Briefly revise from previous grades. **IT** is all about using computers (and other computing devices and tools) such as smart watches, etc. to process large amounts of data rapidly, to provide information. For example, the school processes learners' marks (data) to provide information (report with averages per subject and grade to provide context and enable comparison) Extend to data, information, decisions and simulations: **Data and Information** In our world today, there's a lot of data and information around us – from text and pictures to videos and sounds. IT helps people manage, organize, and make sense of this information. It's like sorting your toys into different boxes to keep them neat and easy to find. In the world of IT, **data** is like raw material – it's a bunch of facts, numbers, and details that may not make much sense on its own. IT helps one in using various techniques to **process and transform this data** into meaningful **information**. This process involves cleaning, organizing, and analysing the data to uncover patterns, trends, and insights. Programming is used to perform these tasks. For example, imagine you have a list of sales numbers – IT can help turn that list into a graph that shows which products are selling the most, making it easier for business decisions, e.g. which products to get in high quantities.  **Mathematics and statistics** are also crucial tools in IT. Once data is processed into meaningful information, these tools help make sense of it. By applying mathematical and statistical techniques (e.g. calculating averages), one can identify trends, predict future outcomes, and **make informed decisions**. For instance, a company might analyse customer purchase history to predict which products will be in high demand during certain seasons. This information guides their decisions about production, marketing, and inventory management.	Learners need to understand • What data is • What information is • Difference between data and information • How data result in information that leads to decision making and simulations **Data** is raw facts without context or meaning **Information** is processed or organised data (data turned into something useful or meaningful) **Decision** can be based on the information one has, e.g. making a choice or picking a course of action. Done with D.3 and D.7 Learners need to understand that • Information Technology (IT) specifically refers to the use of computers and software to manage, process and represent data and information. • The purpose of Information Technology (IT) is to use computers, software, and other technology tools to manage, process, store, and present information in various contexts.

Content (Grade 7 / Term 1)	Notes/Examples
Simulations are like digital experiments that model real-world situations. They are widely used in IT for testing, analysis, and decision-making. Programming allows one to create computer-based simulations that **mimic how systems or processes behave in different scenarios**. For example, in the field of healthcare, simulations can help doctors and researchers study the effects of certain treatments on a virtual patient before applying them in real life. Simulations provide insights into how changes might impact a complex system without implementing those changes. **Example activity – Entrepreneur's Day** In pairs, for your entrepreneur's day at school, your friend and you will have a stall where you could sell something or provide a service for money. • Gather data to determine what you will sell (e.g., ask friends what they want/would like to buy/what were the best sellers the previous time/is it winter or summer (e.g. selling soup vs ice cream) • Decide and plan (you can simulate your plan on paper or using information technology) • Explain how the data you gathered led to information that helped you to come to a decision	
D.3 Demonstrate an understanding of the concept of a computing device.	**Link to D.1, D.2 and D.7 and C.2 and C.3**
Example activity 1 – An exploration of what a computer is and what it can do Distinguish between computers and robots. (When is a robot a computer? When is a computer a robot?) A **computer** is primarily used for data processing and information storage, while a robot is designed for physical action and interaction with its environment 3. Computers are more general-purpose machines, whereas robots are often designed to carry out specific tasks or functions 3. A **robot** is a machine capable of manipulating or navigating its environment, and a computer is not. For example, a robot at a car assembly plant assists in building a car by grabbing parts and welding them onto a car frame. A computer helps track and control the assembly but cannot make physical changes to a car. A **computer** becomes a robot when it controls physical actions or movements in the real world, while a robot is considered a computer when its primary function is computation or data processing, even if it doesn't have a physical body or actions. The distinction depends on the focus and application of the technology involved. Identify computing devices in real-world scenarios and explain why these can be named computing devices. Understand the concepts of input, output, and storage and how computing devices process data and produce results. Provide some appropriate examples of input and output devices and storage (keyboard, mouse, remote control for TV, screen, touchscreen, hard disk). Extend on the concepts of processing, and memory, and incorporate the concept of storage and RAM. **CPU** (**Central Processing Unit)** is the main component that controls most of the operations in computer. It fetches data and program instructions from RAM (input), processes it (execution) and then sends back the results. **RAM (Random Access Memory**) is form of data storage that stores data temporarily. Data that the CPU requires are generally fetched from the hard disk (permanent storage) and placed in RAM to be fetched by the CPU if required. The content of RAM is changed regularly to hold the data that is most likely needed by 	Done with D.7 Learners need to understand • In the context of information technology, a computer is an electronic device designed to process, store, and retrieve data through various programs and applications. • A computer accepts input, processes a set of instructions (block-based program), provides output and stores data and programs. • A computing device can receive data, follow and interpret instructions and produce output/result or render an outcome • Distinguish between computers and robots (When is a computer a robot, when is a robot a computer Learner needs to have a simple understanding of the role of • the CPU and RAM in terms of processing • the hard disk/SSD in terms of storage • input and output devices
D.7 Present a basic understanding of the concept of input processing and output.	**Link to D.3 and C.1, C.2 and C.3**

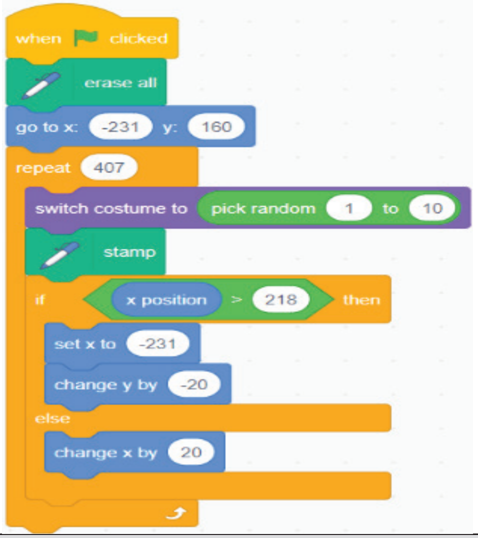
Content (Grade 7 / Term 1)	Notes/Examples
A computer is a programmable device that stores, retrieves and processes data **Input →Processing → Output** **Storage** RAM for temporary (primary) storage SSD /SD card/ flash drive for secondary/permanent storage  	**D.3 and D.7 are done together** Revise and extend from previous grades Learners need to understand what a computer is – general model • Types of computers (PCs (desktop, All-in-one, laptop, hybrid), tablet, smart phone, wearable, server, embedded) • Concept of input → processing → output • Name input and output devices • Name storage devices • Name components involved in processing (CPU and RAM) and their function (simple explanation)
D.10 Demonstrate a basic proficiency in the application of digital skills.	**Link to C.2 and C.3**
Revise and extend from previous grades **File management** Filing is arranging and storing documents in an ordered manner so that they could be found easily when needed.   https://www.youtube.com/watch?v=k-EID5_2D9U	Revise simple file management from previous grades https://www.youtube.com/watch?v=k-EID5_2D9U Learners need to understand • What a file is • What a folder is • The difference between a folder and a file • File naming conventions • Purpose of the file extension • File path • Common types of files (those they are working with) • Folder hierarchy

3.1.2 Term 2

Content (Grade 7 / Term 2)	Notes/Examples
Coding	
C.1 Apply computational thinking (CT) skills to develop a set of logical instructions to solve a problem.	**Link to C.2-C.7, R.5-R.7, D.8-D.9**
Example activity Study the pattern in the following diagram: Identifying the number of rows and columns in the sequence helps to see the pattern: 1 row/column → 1 small square blocks 2 rows/columns → 4 small square blocks 5 rows/columns → 25 small square blocks The pattern then helps us to make **predictions**, that is to **predict** that the next item (after 5 rows/columns) would be: 6 rows/columns with 36 small square blocks followed by 7 rows columns with 49 small square blocks, etc. The pattern also helps us to **generalise**, that is to see a relationship that enables us to devise a symbolic representation of the pattern that will always be true. In this instance there is a relationship between the number of rows/columns and the number of small square blocks in the figure that one can represent in symbolic format as: **n** rows/columns → **n^2** number of small square blocks If one therefore wants to know how many small square blocks the 20th item would have, one could simply use the generalised formula: 20 rows/columns → 20^2 small blocks, that is 400 small square blocks.  Figure: 1 2 3 4 5 n Tiles 1 4 9 16 25 n^2 Learners now write an algorithm that will request the user to enter the number of rows, then display the number of square blocks/tiles.	**Done with C.6 and C.7** Learners need to understand that, in terms of programming, **computational thinking** is an approach to solving problems using concepts and ideas from computer science and expressing solutions to those problems so that they can be executed by a computer. **Computational thinking** involves: • abstraction (focusing on the main ideas) • breaking down a problem into smaller parts (decomposition), • looking for patterns (pattern recognition and generalisation), • developing a step-by-step solution (algorithm). When one uses **computational thinking** to solve a problem, it helps one in developing an **algorithm** – a step-by-step series of instructions. Whether it is a small task like scheduling meetings or a large task like mapping the planet, the ability to develop and describe **algorithms** is crucial to the problem-solving process based on **computational thinking**. To **generalise** is to look at specific cases; identify a pattern or relationship that will always be true; then represent the pattern in symbolic format.
C.2 Present a simple coding solution using symbolic or written statements representing sequences of commands, single repetition, and conditional constructs.	**Link to C.1, C.3, C.4 and C.5**
Example activity 1 – Create a pattern Use your experience of drawing a square in earlier grades to develop code that will draw the pattern on the right: **Hint:** The number of moves must increase with each layer For this pattern, the loop executed 50 times  **Example activity 2 – Draw a similar pattern** See if you can change your code from example activity 1 to draw triangle pattern instead of a square (If your program in activity 1 worked correctly, you only need to make one change to draw the same pattern using a triangle) **Example activity 3 – Draw more patterns** Play around with the code and see how many patterns you can draw.	Leaners use experience from activities done previously, e.g. drawing a square, to draw patterns.

<table>
<tr><th>Content (Grade 7 / Term 2)</th><th>Notes/Examples</th></tr>
<tr><td>C.3 Interpret and execute a given symbolic or written set of commands</td><td>Link to C.1, C.2 and C.4 and link to D.10</td></tr>
<tr><td>

Example activity 1 – Revise loop with IF…ELSE and stamp
Study the code on the right
Explain what the program does
Run the program and see if you predicted correctly



```
when ⚑ clicked
erase all
go to x: -231 y: 160
repeat 407
    switch costume to pick random 1 to 10
    stamp
    if x position > 218 then
        set x to -231
        change y by -20
    else
        change x by 20
```

</td><td>

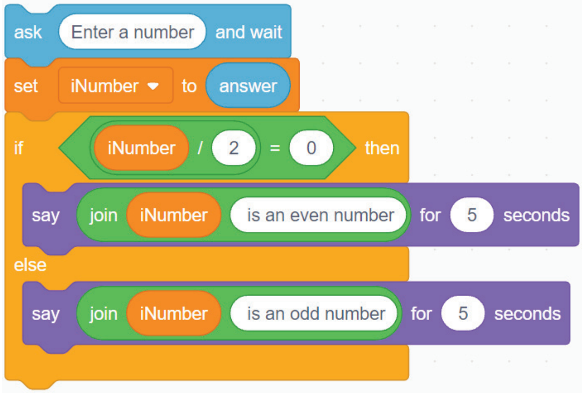
Incorporates C1, C2, C3, C4, C5, C6 and C7

Note:
Literature suggests that it is important that learners must also read and explain in plain language (their own words) what the code does.
This type of activities should be done unplugged (pen-and-paper) and only implemented after learners explained the results. Many of these types of exercises are necessary to ground concepts, skills and understanding of algorithms and coding.

Note:
While learners should be able to describe what each line (block) of code does, (describing a code segment line-by-line/block-by-block) it is very important that learners explain the overall purpose of the code (what program does), i.e. what the program does/the purpose of the program is.

</td></tr>
<tr><td>C.4 Debug a given symbolic or written set of instructions</td><td>Link to C.1, C.2 and C.3 as well as R.6 and D.10</td></tr>
<tr><td>

Example activity – Test if a number is odd or even
The following code is supposed to determine if a number entered is odd or even. However, it does not work
Find the error and correct the code



```
ask Enter a number and wait
set iNumber to answer
if iNumber / 2 = 0 then
    say join iNumber is an even number for 5 seconds
else
    say join iNumber is an odd number for 5 seconds
```

Note

When ***sequencing***, one learns to understand the order of things and about patterns and relationships. By learning to sequence, we develop the ability to understand and arrange purposeful patterns of actions, behaviours, ideas, or thoughts that supports the logical sequencing of coding instructions.

</td><td>

Note:
Learners need to be exposed to a wide variety of coding problems. Typically, at this stage, problems could require learners to

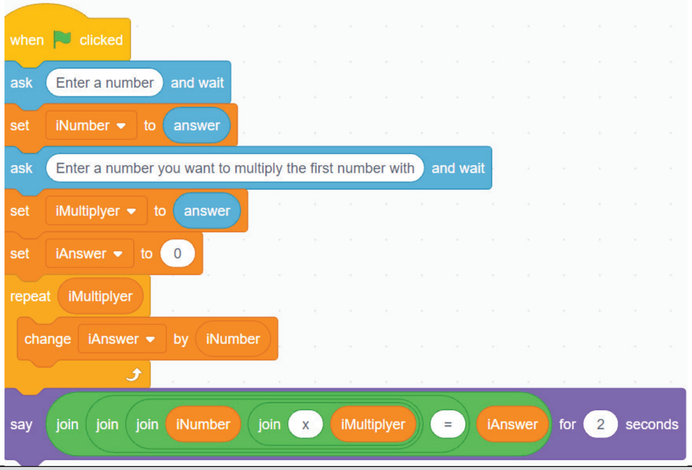
- read code and explain what it does.
- work through (trace) / act out code (physically or simulated) / using pen-and-paper to determine the output or the correctness.
- provide missing code instructions (code instructions are provided with some instructions or code elements missing) that learners need to complete.
- translate verbal/written instructions (algorithm) to code (e.g. write block-based code for a list of symbolic (e.g. arrows)/written instructions)).
- Add additional functionality to code provided/and existing program
- debug an algorithm or block-based program (find the bug, describe the bug and correct it).
- develop a solution/algorithm (code instructions) based on a given problem or for an open-ended problem through planning, implementing, testing and debugging.

</td></tr>
</table>

Content (Grade 7 / Term 2)	Notes/Examples
C.5 Evaluate a given solution towards potential improvement.	

Example activity 1 – Evaluate code and improve code

The snowman wrote a program to multiply two numbers.
It works well; however, it seems cumbersome.
Work through the code and explain how it works.
Evaluate the code and improve the code to be more efficient.



```
when [flag] clicked
ask (Enter a number) and wait
set iNumber ▾ to (answer)
ask (Enter a number you want to multiply the first number with) and wait
set iMultiplyer ▾ to (answer)
set iAnswer ▾ to (0)
repeat (iMultiplyer)
    change iAnswer ▾ by (iNumber)
↻
say (join (join (join (iNumber) (join (x) (iMultiplyer))) (=)) (iAnswer)) for (2) seconds
```

Notes/Examples:

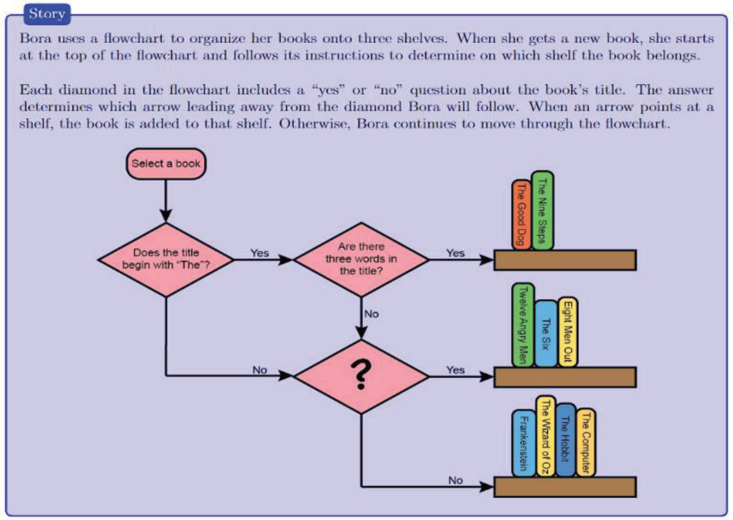
Leaners need to distinguish between different correct solutions in terms of efficiency.

Note:
Cognisance about teaching variables to novices is important as many novices form misconceptions (incorrect mental models) of programming concepts, of which misconceptions about variables are common.

C.6 Recognise and interpret patterns in symbolic sets of data or visualisations.	**Link to C.1 and C.2 and C.7**

Example activity

Read the following story and answer the question that follows:
From: *2020 Beaver Computing Challenge (Grade 7 & 8), University of Waterloo, The Center for Education in Mathematics and Computing* 2020BCCContest7_8.pdf (uwaterloo.ca)



Story

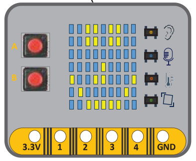
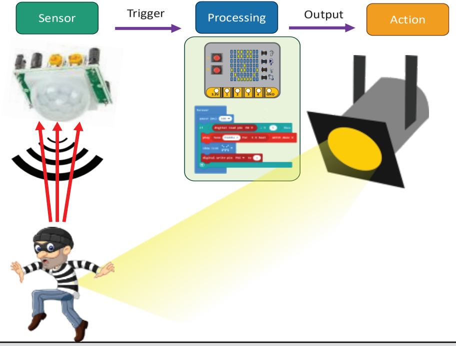
Bora uses a flowchart to organize her books onto three shelves. When she gets a new book, she starts at the top of the flowchart and follows its instructions to determine on which shelf the book belongs.

Each diamond in the flowchart includes a "yes" or "no" question about the book's title. The answer determines which arrow leading away from the diamond Bora will follow. When an arrow points at a shelf, the book is added to that shelf. Otherwise, Bora continues to move through the flowchart.

Question

If Bora's books end up on the shelves as shown, which of the following questions could have appeared in the diamond marked with a question mark (?) in the flowchart?

(A) Does the title include the word "Men"?
(B) Are there fewer than four words in the title?
(C) Is the letter "i" in the title?
(D) Does the title include a number?

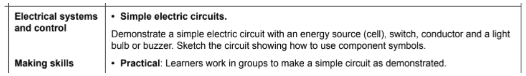
<table>
<tr><th>Content (Grade 7 / Term 2)</th><th>Notes/Examples</th></tr>
<tr><td>C.7 Create or complete a pattern to represent a data set</td><td>Link to C.1, C.2 and C.6</td></tr>
<tr><td>Done with C.1</td><td>C.1, C.6 and C.7 done together</td></tr>
<tr><td>Robotics</td><td></td></tr>
<tr><td>R.3 Outline the different components of a robot</td><td>Link to R.5 – R.7</td></tr>
<tr><td>The learners are introduced to the concept of a microcontroller as a processor.
<table>
<tr><th>Component</th><th>Basic definition</th><th>Purpose</th><th>Practical application example –
May be used to:</th></tr>
<tr><td>Processor (Microcontroller)

 </td><td>A compact integrated circuit that contains a processor, memory, and input/output peripherals, designed to execute specific tasks or control various components within an electronic system.</td><td>The primary purpose of a processor is to execute instructions and perform calculations, making it capable of processing data and controlling the overall operation of the device.
Used as the "brain" of a robot to process information from sensors, make decisions, and control the robot's actions.</td><td>Microcontrollers are at the heart of many robotics projects. They receive data from sensors, perform calculations, and send commands to motors, actuators, and other components to coordinate the robot's behaviour. They also often provide connectivity options for programming and remote control, making them essential for creating intelligent and interactive robots.</td></tr>
</table>
Processor concepts – basic outline of definition and purpose
- Definition of a Processor: Definition: A processor, also known as a central processing unit (CPU), is the brain of a computer or electronic device.
- Purpose: The primary purpose of a processor is to execute instructions and perform calculations, making it capable of processing data and controlling the overall operation of the device.


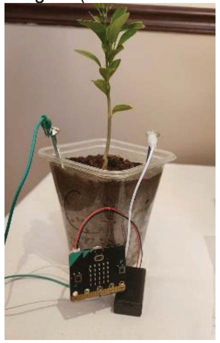
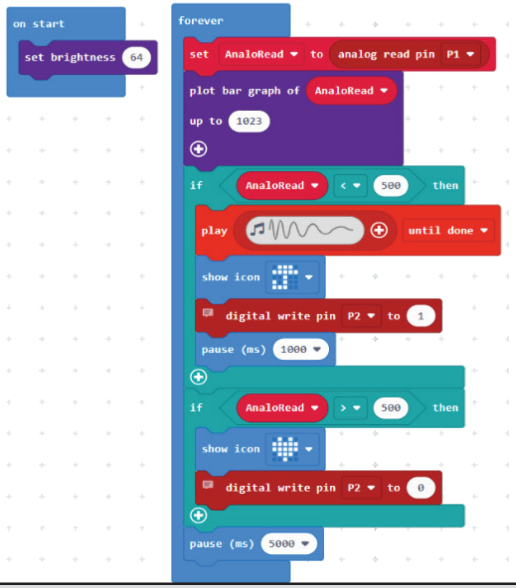
Outline the basic operations and function of a sensor
Outline the basic purpose of a sensor as an input mechanism, which could be processed to trigger an action. Also reemphasise that the processor will be used to control the overall mechanism and operation of the device. The learners need to understand that the code is uploaded and that the device execute the instructions. The processor can control various components such as sensors and actuators through code.</td>
<td>Done in relation to R.5 – R.7 (can be done with R.5 when required).

Extend components of a robot to what is required to build robotic artefacts in Grade 7.

The learners should describe the processor conceptually in terms of a
- Basic definition
- Purpose, and
- how it could be used in simple physical computing and robotics projects.

Robotics concepts are applied in designing and building robots.

Learners need present a simple definition of a processor (microcontroller)</td></tr>
<tr><td>R.4 – Present an understanding of how robots affect the world.</td><td></td></tr>
<tr><td>Example activity - Design a simple task for a hypothetical robot and present it to the class.
The learners will use their design thinking skills and creativity to develop a basic task suitable for a robot to perform, explain its purpose, and demonstrate how it could positively impact people's lives, including ethic</td><td>The learners understanding and be able to:
- differentiate between various types of robots,
- describe the impact of robots on human lives.
- briefly discuss the advantages and disadvantages of using robots, including increased productivity, convenience, and potential job displacement.</td></tr>
<tr><td>R.5 Design a simple product (artefact) based on a set of design specifications.</td><td>R.5 and R.6 are done together</td></tr>
<tr><td>R.6 Mimic the operations of a robot</td><td rowspan="2">Links with Technology Grade 7

<table>
<tr><td>Electrical systems and control</td><td>• Simple electric circuits.
Demonstrate a simple electric circuit with an energy source (cell), switch, conductor and a light bulb or buzzer. Sketch the circuit showing how to use component symbols.</td></tr>
<tr><td>Making skills</td><td>• Practical: Learners work in groups to make a simple circuit as demonstrated.</td></tr>
</table></td></tr>
<tr><td>Example activity - Soil moisture sensor artefact
The learners design a simple soil moisture sensor artefact using a plant and two simple probes (e.g., nails, screws) and an LED + MicroBit. The LED lights up when the soil moisture is low, and a message icon is displayed on the Micro: bit.</td></tr>
</table>

Content (Grade 7 / Term 2)	Notes/Examples
Soil moisture with simple probes Stage 1 (Initial version without LED)  Stage 2 (Version with LED)  	

```
on start
    set brightness 64
forever
    set AnaloRead ▾ to analog read pin P1 ▾
    plot bar graph of AnaloRead ▾
    up to 1023
    ⊕
    if AnaloRead ▾ < ▾ 500 then
        play ⊕ until done ▾
        show icon ▾
        digital write pin P2 ▾ to 1
        pause (ms) 1000 ▾
    ⊕
    if AnaloRead ▾ > ▾ 500 then
        show icon ▾
        digital write pin P2 ▾ to 0
    ⊕
    pause (ms) 5000 ▾
```

Content (Grade 7 / Term 2)	Notes/Examples
R.7 Create, test, and execute a set of robotic instructions.	Link to R.6
Example activity 1 - Blind roll-a-dice The purpose of this program is to create a dice that can be used by visually impaired people. The microcontroller must display a random number between 1 and 6 when shaken and must make a number of beep sounds according to the number that was is displayed. For example: If the dice lands on 3, make 3 beeps. • Shake the microcontroller to display a random number between 1 and 6 that can be used as a dice. • Use a loop to play the number of beep sounds that correspond with the number. **Example activity 2 - Compatible-group-calculator** Divide learners into pairs (pair programming). Each pair creates a program for the microcontroller to "read" each group member's personality to decide if learner A and learner B will be able to work well together. (This is just a game, and it cannot be done in reality). Learners play the game with each other until they all end up in compatible pairs. The compatible group calculator must work as follows: • Learner A presses Button A for a while (long press) and the microcontroller generates a random number between 0 and 9 and displays it on the screen. • Learner B presses Button B for a while (long press) and the microcontroller generates a random number between 0 and 9 and displays it on the screen. • If the difference between the two numbers is less than 10, display a smiley face to indicate that the pair is "compatible", else display a "X" to indicate that the pair is not compatible. If the pair is not compatible, the learners play the game with other learners to find a peer that will be compatible to work with, according to the compatible group calculator.	Code the microcontroller to achieve a specific outcome.
Digital Concepts	
D.2 Recognise that he or she is living as citizens in a digital world.	**Link to D.4**
Learners present an understanding that the digital world is all around us. Identify the ethical issues associated with the use of technology, including privacy, security, and social responsibility. Present an understanding of the dangers of the online environment.	Can be integrated with other aspects of digital concepts, e.g. Internet Learners need to understand the concept of

Content (Grade 7 / Term 2)	**Notes/Examples**
Example activity 1: Make a list of ten differences and similarities between your online and offline identity. **Example activity 2**: Have a class discussion about safety measures to consider when being/working/engaging online.	• Digital citizenship (ethical issues and social responsibility) • Privacy and security when online
D.3 Demonstrate an understanding of the concept of a computing device.	**Link to D.1, D.5, D.7**
Extend to how devices in a network (using a modem, router) connect to each other to communicate. Basic understanding (Refer to D.5 and D.7) A modem connects your home to the internet by translating data into a format that can travel over the internet, while a router manages and directs the traffic between devices in your home network and connects them to the internet. Together, they ensure that your devices can communicate with each other and with the vast world of the internet.	Do with D,5 and D.7 Learners need to understand (basic) what a • Modem is and what it does • Router is and what it does
D.4 Identify the common uses of ICT in the real world	**Link to D.1 and D.3 and D.5**
Briefly revise what an ICT system is and its basic components **Example activity – How ICT provides a fantastic way to stay connected and feel close even when you're physically far apart** Divide learners into small groups and provide the following example of ICT. **Example of ICT:** Imagine you have a friend or a family member who lives far away, and you want to see and talk to them. Instead of traveling a long distance to meet them in person, you can use ICT to connect with them through an online video call. Here's how it works. **Devices**: You and your friend each need a device with a camera and microphone, like a smartphone, tablet, or computer. These devices are part of the "information" aspect of ICT **Internet Connection**: Both of you need to be connected to the internet. The internet allows you to send and receive information quickly, including video and audio **Application (Software)**: You use a special application (app) for video calling. These apps are designed to handle video and audio communication over the internet. You and your friend need to use of the same app. **Initiating the Call**: You open the app, find your friend's contact, and start the video call. The app uses your device's camera to capture your video and microphone to capture your voice **Transmission**: The app converts your video and audio into digital data (binary) and sends the data over the internet to your friend's device **Receiving the Call**: Your friend's app receives the data, decodes it, and displays your video on their screen. Similarly, they can talk, and their voice gets sent back to you. **Real-Time Interaction**: Now you and your friend can see and hear each other in real time, almost as if you're in the same room. You can chat, share stories, and even show each other things using your cameras Each group discusses the example and come up with their own example of ICT, then report back on their example. The example must show, like the one above how ICT combines information processing (using the app and devices) with communication (network, video and audio) to bridge the distance gap between, e.g. people.	Do with D.1, D.2 and D.6 as well as D.3 and D.7 Learners need to understand: • What IT is and the purpose of IT • What ICT is and the purpose of ICT • Basic components of an ICT system • Provide example of both IT and ICT While IT focuses on the technical aspects of technology, ICT looks at how technology can be used for both information processing and communication One can think of IT as the field that deals with all things related to computers and technology, while ICT expands that to include technology's role in communication as well. ICT is a more comprehensive term that goes beyond just information processing and includes communication aspects. While ICT includes everything under IT, it also encompasses the use of technology for communication purposes.
D.5 Differentiate between the components of an ICT system	**Link to D.3, D.5 and D.7**

Reinforce and extend from the previous grades and terms
Remind learners that ICT is an umbrella term that includes communication devices and systems (devices connected in a point-of-sales (POS) system, cellular network). such as a cellphone network using towers and point-of-sales (POS) using tills, barcode readers, card machines
Relate the concept of computers to that of an ICT tool.
In modern computing, computing devices interact with each other using different methods.
Networks allow computing devices to communicate with each other and share data and information.

Example activity – PC communicating to the Internet
Explain to learners that to connect a computer to the internet, one needs a modem
• Connecting one computer to the Internet, only requires a modem
    • Computer linked directly to modem
    • Modem allows internet connection using a wired or wireless connection.
	D.3 and D.7 are done together
	Learners need to
	• relate a computing device to that of an ICT tool
	• know (at a basic level) what a network is
	• know that a network enables devices to communicate with each other using different methods (wired and wireless)
	• name examples of daily life situations where devices need to communicate, like sending a message from a phone to a friend's phone or printing a document from a computer to a printer
	• additional computing devices are required to enable communication (modem and router)

Content (Grade 7 / Term 2)

- Connecting more than one computer to the Internet also requires a router, e.g. Wi-Fi router / modem-router
- Devices connect to modem-router via wired (cables) or wireless (Wi-Fi, cellular)

Modems and routers are computing devices that allow computers to connect to the internet to communicate with other computers worldwide.
https://youtu.be/mYXuCeawhm8

Notes/Examples

Modem – a computing device that allows other computing devices access to the internet using cables, e.g. fibre (wired connection)

Router – a computing device that allows multiple computers to connect to the same network and routes the data between the devices

Routers generally also includes modems, which means the computers connected to the router can also access the internet.

D.7 Present a basic understanding of the concept of input processing and output.

Example activity – Networks and communication
Explain what a network is (could use an analogy such as roads with cars connecting homes), the importance of networks such as the internet, social media and communication

Link to C.2- C.5 and D.10

Learners must

- Distinguish between input through instructions that are executed and results in action and output as a form of communication from the device.
- Suggest appropriate input, output and communication devices for different applications, based on their specifications and features.
- Understand the concept of sharing data using a network.
- Identify connectivity in devices (Bluetooth, Wi-Fi, cellular, wire (cables)).

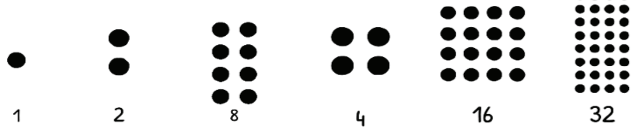
D.8 Interpret a pattern to represent or communicate a message or image

We use the decimal number system that can be described as ones, tens, hundreds, thousands, etc. and which can be represented as follows:

One	Tens	Hundreds	Thousands	Ten thousand	Hundred thousand
1	10	100	1000	10000	100000
1	10 x 1	10 x 10	10 x 10 x 10	10 x 10 x 10 x 10	10 x 10 x 10 x 10 x 10
10^0	10^1	10^2	10^3	10^4	10^5

The number 4567 can be written as 4000 + 500 + 60 + 7 which can be represented as $4x10^3 + 5x10^2 + 6x101 + 7x10^0$
The decimal number system works in groups of 10
Introduction to binary numbers.
Binary is a coding system using the binary digits 0 and 1 to represent a letter, digit, or other character in a computer or other electronic device.
Explain that all data (text, music, and images) are represented using only two digits: 0 and 1.

Example activity 1: Use cards to introduce the binary system.
The decimal number system works in groups of 10s, but the binary number system works in groups of 2s
With the decimal system one uses: ten symbols (0, 1, 2, 3, 4, 5, 6, 7, 8, 9), but with the binary system one only uses two symbols, 0 and 1



Link to squares/power of decimal numbers: 2^0, 2^1, $2^{2,}$ 2^3, 2^4, 2^5

Link to C.1, D.2 and R.5 – R.7

Introduce the binary number system (link to decimal number system)
Learners need to be able to

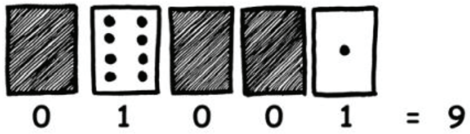
- Know what the binary number system is
- Know what the binary number system is used for
- do simple conversions between decimal numbers and binary code

	32	16	8	4	2	1
0	0	0	0	0	0	0
1	0	0	0	0	0	1
2	0	0	0	0	1	0
3	0	0	0	0	1	1
4	0	0	0	1	0	0
5	0	0	0	1	0	1
6	0	0	0	1	1	0
7	0	0	0	1	1	1
8	0	0	1	0	0	0
9	0	0	1	0	0	1
10	0	0	0	0	1	1

Content (Grade 7 / Term 2)

Write the numbers 0, 1, 2, 3 on the left side of the board underneath a horizontal line. See the example on the right.
Ask the learners some questions to get a further three numbers between 0 and 64 and add these to the number column. Ask for example:

- How many learners are in the classroom?
- How many learners have siblings?
- How many learners have a name starting with a particular letter?



Ask the learners if they think we can write these numbers using just zeros and ones.
Give a volunteer (person 1) a card with one dot on one side and a black background on the other and write 1 on the far right of the board above the horizontal line.
Explain that 1 means on and 0 means off, so the black side represents 0. This way, we can represent the numbers 0 and 1.
Give another volunteer (person 2) a card with two dots. Place this learner on the left of the first one and write 2 to the left of 1 on the board.
Ask them how they can, together, represent the number two. What about three? Ask the class for their ideas. Ask them to flip the cards to the on/off positions to help the discussion.
Extend the game by giving 6 learners a card to represent different number. Some will be showing, some will not be showing. When a binary number card is not showing, it is represented by a zero. When it is showing, it is represented by a one.
Extend to larger numbers.

<u>Example activity 2</u>: Crack the secret code
Give learners a secret code to crack.
Each letter is represented by a number.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	23	23	24	25	26
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Convert the binary representation below to determine the decimal number and look up the letter. Find the secret code.

16	8	4	2	1	Letter
■	■	■			
■					
■	■		■	■	
■		■	■		
■				■	
■	■				

Let learners create their own secret code and let their friends crack that.

<u>Example activity 3</u>:
Let learners create a clock showing the current time using binary code.
Example on the right

Notes/Examples

Ask the learners if they own a computer or a smartphone.
How do these machines store and share information?
Learners need to understand that:

- Binary is a numbering system that uses only two digits: 0 and 1. This contrasts with our familiar decimal system, which uses ten digits (0-9).
- Representation of Information: In the binary system, each digit (0 or 1) is called a "bit" (short for binary digit). Bits are the basic units of information storage and processing in computers. They can represent two distinct states, often referred to as "off" (0) and "on" (1), or "false" and "true'.

Note:
Binary is a fundamental aspect of digital systems such as computers and microcontrollers.
In most digital systems, information is ultimately represented and processed using binary digits (0 and 1). It's the simplest and most common way to represent information digitally because it's easy for electronic circuits to work with two distinct states.

Let them think of a digital system as a series of tiny switches that can be either on or off, like a light switch.
In the digital world, everything is represented using these on-off switches. Each switch is like a tiny piece of information, called a "bit." When you put many bits together, you can represent more complex things, like numbers, letters, pictures, and sounds.

D.10 Demonstrate a basic proficiency in the application of digital skills.

Example activity - Paint
Learners use software such as Paint to create sprites and backgrounds for importing into their block-based programs

Files and folders - File and folder operations

Link to C.2, C.3, C.4 and C.5 and D.3 and D.7

Revise Paint
File management
Filing is arranging and storing documents in an ordered manner so that they could be found easily when needed.

Content (Grade 7 / Term 2)	Notes/Examples
What one can do with files  What can one do with folders?  **Example activity – File and Folder Management** When learners work on the computer (coding), they engage in file and folders operations	Learners need to • Create and name a folder and file • Rename a folder and file • Delete, copy and move a folder (difference between copy and move) • Save and Save a copy (and the difference) • Move a folder and file (difference between copy and move) • Delete a folder and file • Concept of recycle bin

3.1.3 Term 3

Content (Grade 7 / Term 3) | **Notes/Examples**

Coding

C.1 Apply computational thinking (CT) skills to develop a set of logical instructions to solve a problem.

Link to C.2 – C.4 and C.6 and R.1-R.6

Example activity – Use computational thinking to develop an algorithm for the following task

If you need to find a picture of a computer that you want to import into a block-based coding app, you will break down the task into several smaller tasks, that can be completed individually.

The first step is to break the problem into two distinct sub-tasks:

- Find a picture of a computer on your computing device; and
- Import the picture into the block-based coding app.

Now, break down each of the above tasks into even smaller tasks:

- Find a picture of a computer on the Internet
  - Complete these steps
- Insert the picture into the word processing document
  - Complete these steps.

> A **high-level algorithm** includes only the **main** steps/ideas or instructions of the solution and leaves the details until later (**Abstraction to Decomposition**)
> **Refinement** is a process for developing a clear, unambiguous and detailed algorithm by breaking the high-level steps down into detailed steps (gradually adding detail to a **high-level ideas)**, finding patterns, etc. (**Decompose**)

Learners need to:
Identify the important details needed to solve this problem (abstraction)
Break the problem down into smaller logical steps (decomposition)
Explain to learners:
When they need to complete a task and feel overwhelmed, it is important to break it down into sub-tasks. Then solve each sub-task.

C.2 Present a simple coding solution using symbolic or written statements representing sequences of commands, single repetition, and conditional constructs.

Link to C.1 – C.7 and D.8

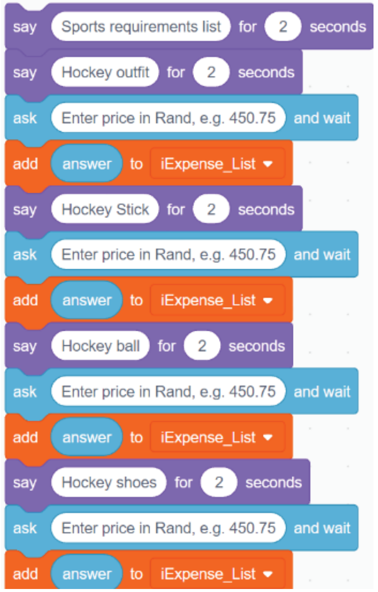
Example activity 1 – Convert a 3-digit binary number to a decimal number

Learners must write code to do the following:

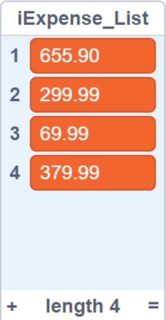
- The user must enter a 3-digit binary number
- The program must convert the 3-digit binary number to the corresponding decimal number.
- The program must display the binary number and the corresponding decimal number.

Example activity 2 – Create an expense list

You will be playing hockey.
Create a list called rExpence_List (under variables)
Write the code on the right, run the code, enter the prices and add the prices to the list.
The list will store the prices entered.



```
say Sports requirements list for 2 seconds
say Hockey outfit for 2 seconds
ask Enter price in Rand, e.g. 450.75 and wait
add answer to iExpense_List
say Hockey Stick for 2 seconds
ask Enter price in Rand, e.g. 450.75 and wait
add answer to iExpense_List
say Hockey ball for 2 seconds
ask Enter price in Rand, e.g. 450.75 and wait
add answer to iExpense_List
say Hockey shoes for 2 seconds
ask Enter price in Rand, e.g. 450.75 and wait
add answer to iExpense_List
```

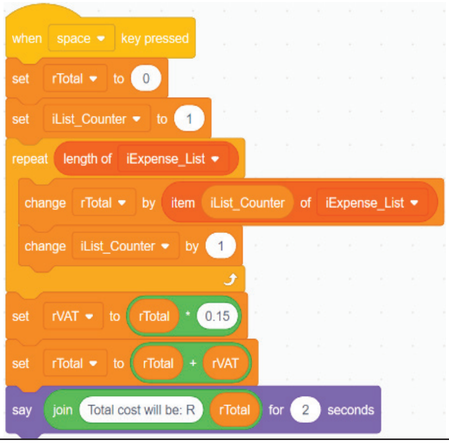


iExpense_List	
1	655.90
2	299.99
3	69.99
4	379.99
+	length 4 =

After the list is created, you need to calculate the total cost when pressing the spacebar

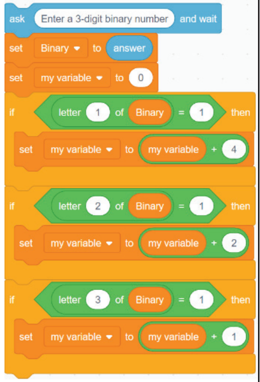
- use the prices in the list to calculate how much money you will need
- then calculate VAT and
- add the VAT to the total:

Add the following code:



```
when space key pressed
set rTotal to 0
set iList_Counter to 1
repeat length of iExpense_List
    change rTotal by item iList_Counter of iExpense_List
    change iList_Counter by 1
set rVAT to rTotal * 0.15
set rTotal to rTotal + rVAT
say join Total cost will be: R rTotal for 2 seconds
```

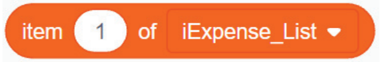
Possible solution for activity 1



```
ask Enter a 3-digit binary number and wait
set Binary to answer
set my variable to 0
if letter 1 of Binary = 1 then
    set my variable to my variable + 4
if letter 2 of Binary = 1 then
    set my variable to my variable + 2
if letter 3 of Binary = 1 then
    set my variable to my variable + 1
```

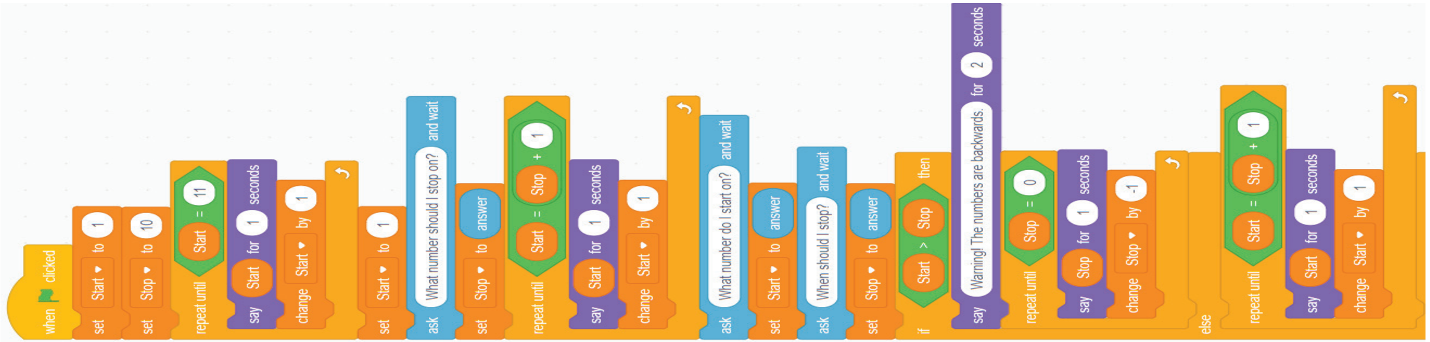
Note:
Lists are data structures that allow programmers to store multiple pieces of information in a single variable.

Ensure that learners understand that the item number refer to a specific item in the list, e.g. 1 refers to the first item, 2 to the second, etc. Therefore, one needs a list counter variable that must change by 1 each time to work through the list.



```
item 1 of iExpense_List
```

Content (Grade 7 / Term 3)	Notes/Examples
C.3 Interpret and execute a given symbolic or written set of commands	**Link to C.1, C.2 and C.4**
Example activity 1 – Revise loops Work through the code below. Explain what the code does. Run the code and check if you predicted correctly	



```
when clicked
set Start to 1
set Stop to 10
repeat until Start = 11
    say Start for 1 seconds
    change Start by 1
set Start to 1
ask What number should I stop on? and wait
set Stop to answer
repeat until Start = Stop + 1
    say Start for 1 seconds
    change Start by 1
ask What number do I start on? and wait
set Start to answer
ask When should I stop? and wait
set Stop to answer
if Start > Stop then
    say Warning! The numbers are backwards. for 2 seconds
    repeat until Stop = 0
        say Stop for 1 seconds
        change Stop by -1
else
    repeat until Start = Stop + 1
        say Start for 1 seconds
        change Start by 1
```

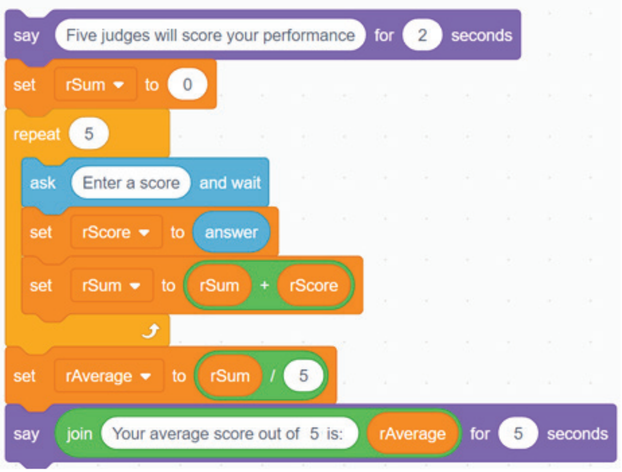
Content (Grade 7 / Term 3)	Notes/Examples
C.4 Debug a given symbolic or written set of instructions	
Example activity – Calculate multiples of a number The program on the right asks the user to enter a number, then ask the user how many multiples of that number it must display. If the user enters 12 and ask for 9 multiples of 12 to be displayed, the output must display 12, 24, 36, 48, 60, 72 Hower, the program contains a bug. Use a trace table to find the bug, correct it and test the code. (Repeat until the program works correctly).	**Note**: When writing code, learners must be encouraged to use trace tables for finding bugs in their programs. They must also be encouraged to persist with debugging until the program works (very few programs run perfectly the first time and generally needs a couple of iterations in terms of debugging) (Solution: iCounter remains 1 – it should be changed by 1 as the last instruction in the loop)

```
when clicked
ask Enter the number of which you want to see multiples of
set iNumber to answer
ask How many multiples must be displayed? and wait
set iNumber_of_Multiples to answer
set iCount to 1
repeat iNumber_of_Multiples
    set iMultiple to iNumber * iCount
    say join Multiple join join iCount of join join iNumber is: iMultiple for 2 seconds
```

Content (Grade 7 / Term 3)	Notes/Examples
C.5 Evaluate a given solution towards potential improvement.	
Example activity – Calculate competition score A robotics competition has five judges that judge learner's robotic artefacts. Each judge assigns a score out of 10. The learner's final mark is the average of the five scores. The following two prob=grams were written to receive the scores from the five judges and calculate the average. Study the programs (use a trace table to work through each program) and explain how each one works. Evaluate both programs and explain which program is the most efficient,	**Note**: Learners need to understand that most problems have more than one solution. However, some solutions are better than others.

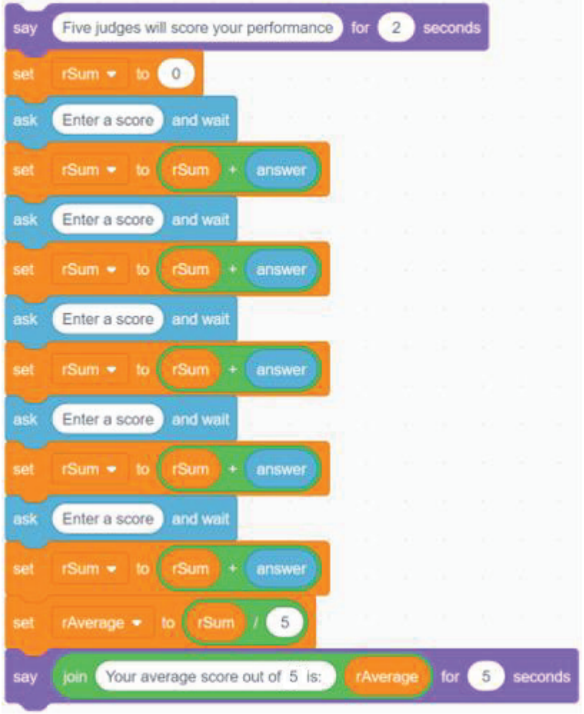
Content (Grade 7 / Term 3)	Notes/Examples

Program 1



```
say [Five judges will score your performance] for (2) seconds
set [rSum] to (0)
repeat (5)
    ask [Enter a score] and wait
    set [rScore] to (answer)
    set [rSum] to ((rSum) + (rScore))
set [rAverage] to ((rSum) / (5))
say (join [Your average score out of 5 is:] (rAverage)) for (5) seconds
```

Program 2

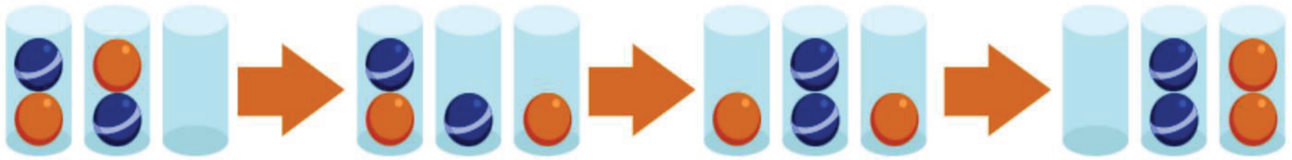


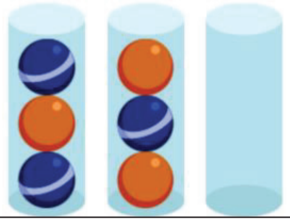
```
say [Five judges will score your performance] for (2) seconds
set [rSum] to (0)
ask [Enter a score] and wait
set [rSum] to ((rSum) + (answer))
ask [Enter a score] and wait
set [rSum] to ((rSum) + (answer))
ask [Enter a score] and wait
set [rSum] to ((rSum) + (answer))
ask [Enter a score] and wait
set [rSum] to ((rSum) + (answer))
ask [Enter a score] and wait
set [rSum] to ((rSum) + (answer))
set [rAverage] to ((rSum) / (5))
say (join [Your average score out of 5 is:] (rAverage)) for (5) seconds
```

C.6 Recognise and interpret patterns in symbolic sets of data or visualisations.	Link to C.7
C.7 Create or complete a pattern to represent a data set	C.6 and C.7 are done together

Example activity – Moving balls following specific rules

Let us play a game. The aim of the game is to move balls from one tube to another so that balls of the same colour are in the same tube, e.g.





The following rules for playing the game apply:

- Rule 1: A ball can be moved to an empty tube.
- Rule 2: When there is a space in a tube, a ball can only be moved on top of a ball of the same colour.
- Rule 3: Only one ball at the top of a tube can be moved at a time.

Given that the balls are moved from tube to tube following the rules of the game: what is the minimum number of moves needed to get the same colour balls in the same tubes for the problem on the right?

2022-TS-Intermediate-Question-Paper.pdf (olympiad.org.za)

Content (Grade 7 / Term 3) | **Notes/Examples**

Robotics

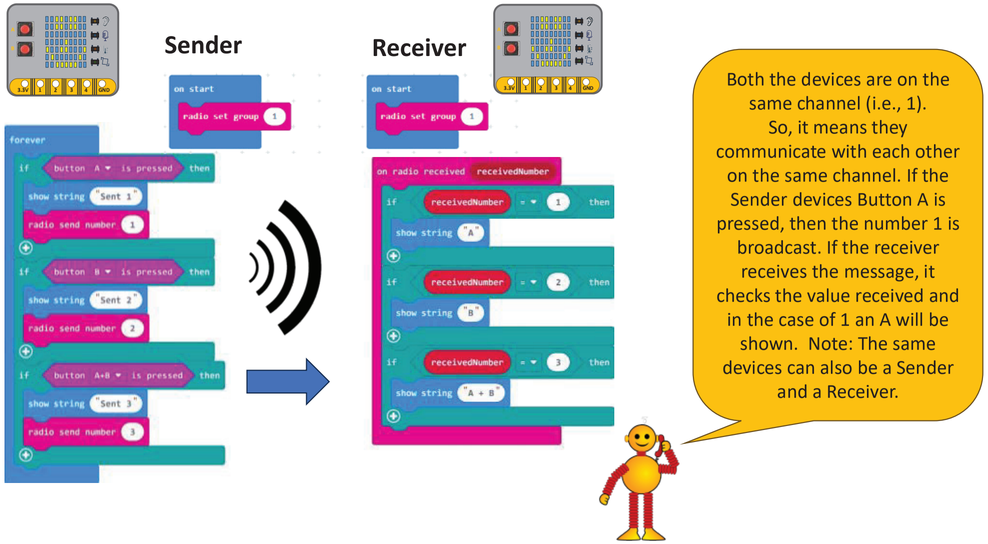
R.3 Outline the different components of a robot

Components to cover

Component	Basic definition	Purpose	Practical application example – May be used to:
Servo 	A type of motor that can precisely control angular position.	Used for precise movement and positioning in robotics, often for tasks such as limb control and object manipulation.	Enables precise movement of robot limbs, grippers, and other mechanical components such as cameras, or gripping mechanisms.
PIR 	A sensor that detects changes in infrared radiation, typically associated with motion.	Used for detecting the presence of moving objects or individuals in the robot's surroundings. (They help robots see if there is something moving nearby. It can detect people or animals and let the robot know if someone is there)	In robotics, PiR sensors are used to detect motion, presence, and create interactive robot behaviour, such as obstacle avoidance. It can also be used to detect motion in the robot's vicinity, enabling security or interactive features.

Basic introduction to Radio Communication (2 Devices)

The learners are introduced to the concept of message passing between two devices. For grade 7 this is done on a very elementary level just to illustrate the concept.



Sender:

```
on start
  radio set group 1

forever
  if button A ▾ is pressed then
    show string "Sent 1"
    radio send number 1
  ⊕
  if button B ▾ is pressed then
    show string "Sent 2"
    radio send number 2
  ⊕
  if button A+B ▾ is pressed then
    show string "Sent 3"
    radio send number 3
  ⊕
```

Receiver:

```
on start
  radio set group 1

on radio received receivedNumber
  if receivedNumber = ▾ 1 then
    show string "A"
  ⊕
  if receivedNumber = ▾ 2 then
    show string "B"
  ⊕
  if receivedNumber = ▾ 3 then
    show string "A + B"
  ⊕
```

Both the devices are on the same channel (i.e., 1). So, it means they communicate with each other on the same channel. If the Sender devices Button A is pressed, then the number 1 is broadcast. If the receiver receives the message, it checks the value received and in the case of 1 an A will be shown. Note: The same devices can also be a Sender and a Receiver.

Also see: https://microbit.org/projects/make-it-code-it/send-a-smile/

Notes/Examples:

The learners are introduced to the concept of a microcontroller as a processor.
Learners need to provide a basic definition and purpose of the following basic electronic components of a robot:

- Servo
- PR Sensor

The learners should describe the processor conceptually in terms of

- Basic definition
- Purpose, and
- How it could be used in simple physical computing and robotics projects.

In grade 8 the learners will be required to present a deeper understanding of the operations of the components.

R.4 Present an understanding of how robots affect the world

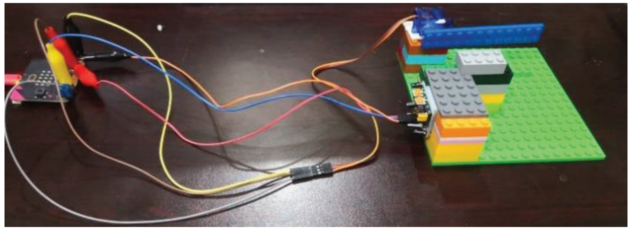
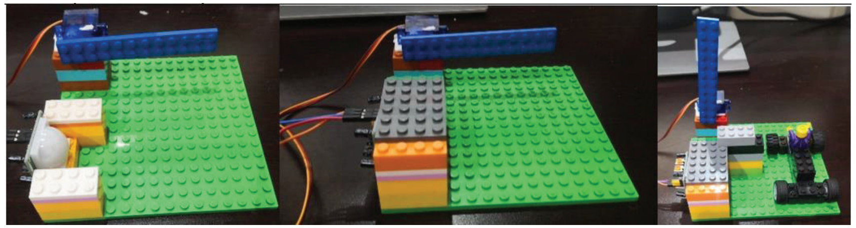
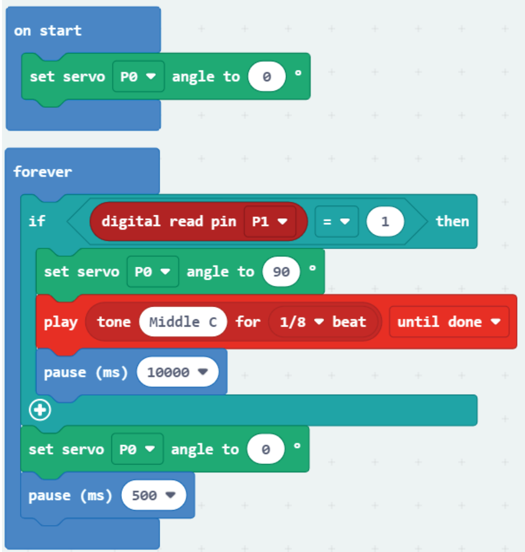
Example activity – Types of robots and how they affect the world

Provide learners with videos to watch, e.g. https://youtu.be/t0br-rADDYw and https://youtu.be/b2GvuPCNhs0 and a KWLS chart.
Learners watch videos and complete the KWLS chart
Now, divide learner into pairs. Each pair identify a field/type of robots, e.g. autonomous robot in the hospitality industry, discuss advantages/disadvantages and design a simple task for the robot to do.

Notes/Examples:

Learners must be able to

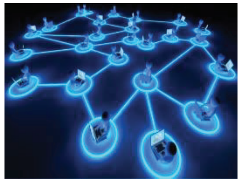
- differentiate between various robots, such as industrial robots, service robots, and educational robots
- describe the positive and negative impacts of robots on human lives.
- discuss the advantages and disadvantages of using robots, including increased productivity, convenience, and potential job displacement.

Content (Grade 7 / Term 3)	**Notes/Examples**
	- design a simple task for a hypothetical robot and present it to the class.
R.5 Design a simple artefact based on a set of design instructions	**R.5 and R.6 are done together**
R.6 Mimic the operations of a robot **Example projects** **Example project 1 - Automatic Door opener using a Servo + Loud sound trigger (Clapping hands opens and closes the door)**  **Example project 2 - Alternate - Automatic Boom Gate using a PIR + Servo**   	**Link to R.5 and R.6 and C.1 – C.7** **Note:** **Project**  The learners must be able to apply the principles of design thinking to develop an artefact with a sensor and actuator, and one additional component. Sensors on the microcontroller may also be used. E.g., Alarm with PIR or Ultrasonic sensor and additional component e.g., External buzzer, LED.

```
on start
  set servo P0 ▾ angle to 0 °

forever
  if digital read pin P1 ▾ = ▾ 1 then
    set servo P0 ▾ angle to 90 °
    play tone Middle C for 1/8 ▾ beat until done ▾
    pause (ms) 10000 ▾
  ⊕
  set servo P0 ▾ angle to 0 °
  pause (ms) 500 ▾
```

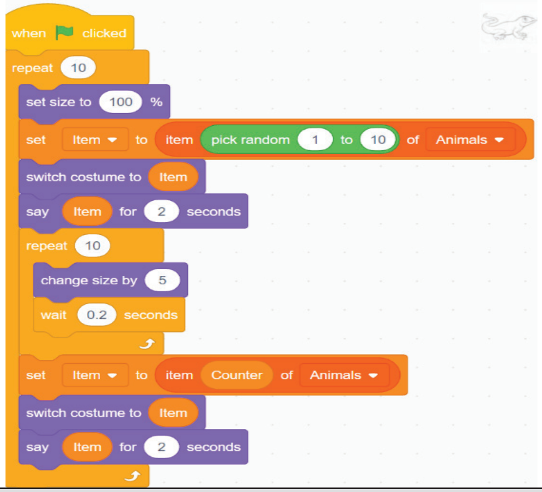
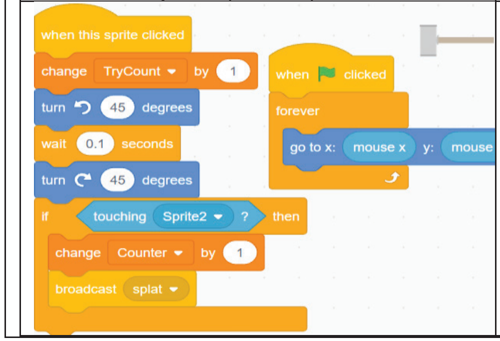
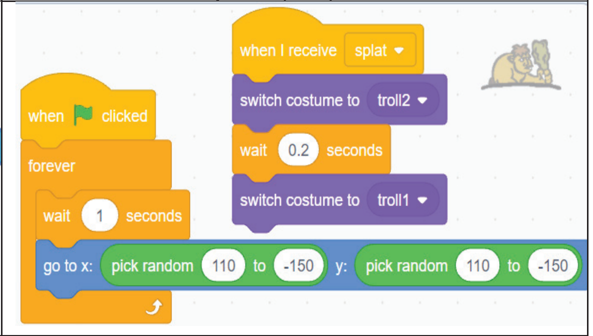
R.7 Create, test and execute a set of robotic instructions	**Link to R.5 and R.6 and C.1 – C.7**
Example activity 1- Pick-a-letter to play “Names, Places, Objects and Animals” Divide learners into groups with 4 learners each. Each group writes code for a game is to let the microcontroller pick a random letter to play the game “names, places, objects and animals”. When a letter is picked a group has 30 seconds to name a person’s name, a place, an object and an animal’s name that starts with	Learners develop a program for the microcontroller to perform a specific task such as playing a game and keeping score

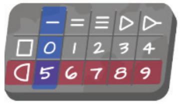
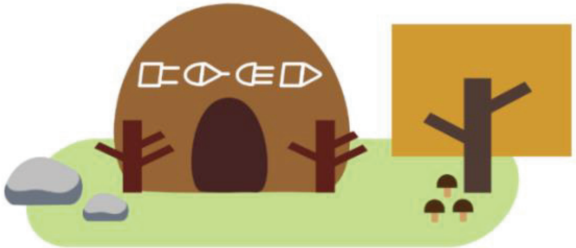
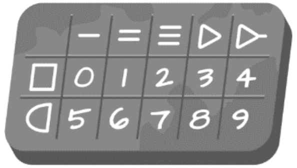
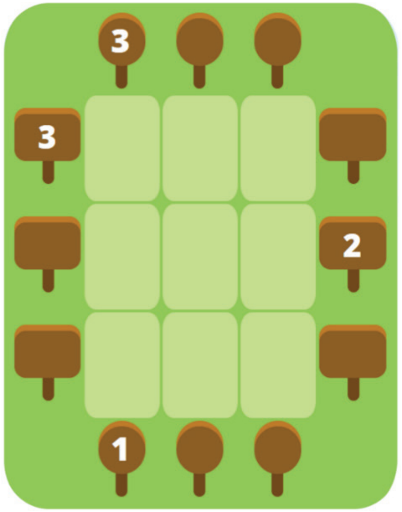
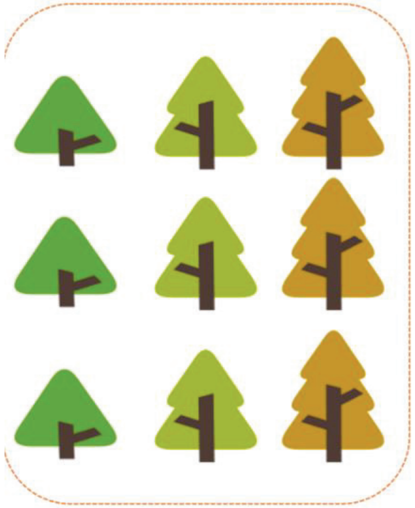
Content (Grade 7 / Term 3)	Notes/Examples
that letter. If the group gets all four answers right, they get four points. If they get three answers right, they get three points. If they get two answers right, they get two points. If they get one answer right, they get one point. After writing the game, learners in the group play in groups of two (2 groups of 2). (Swop group roles below for each round). Group A: Shake the microcontroller to choose a random letter and to start the countdown of 30 seconds. Group B: Must the come up with a person's name, a place, an object and an animal that starts with the letter displayed, within 30 seconds. If they get all four answers right, they get 4 points. If they get three answers right, they get 3 points. If they get two answers right, they get 2 right. If The group gets 1 point if they have one right, 2 points, if they have 2 right, etc. Program the microcontroller to keep the total score (each time add the points that a group got to their previous score). Button A must be pressed to keep the score of group A. Button B must be pressed to keep the score of group B. (Tip: Press a button three times if three points were scored). Press both buttons A and B together to display each group's points and to say who the winner is.	
Digital Concepts	
D.1 Outline the concept of technology and purpose of information technology (IT)	**Link to D.2, D.3 and D.4**
Example activity – Data →Information→Decisions Discuss the following with learners: **Data collection** – Learners imagine data as raw materials, like puzzle pieces spread all over the place - on computers, phones, and the internet. IT professionals help collect these puzzle pieces and put them in one place **Organising data** - Raw data is often messy and unorganized, like a jumble of puzzle pieces. IT professionals, especially programmers, write special instructions (code) to organize these pieces. It's like putting similar pieces together to form a picture **Storing data** - Once the data is organized, it needs a safe place to stay, like a storage room. IT experts build and manage these digital storage rooms, which are called databases **Processing data** - To make sense of data, one needs to do things with it. IT professionals create programs (like computer apps) to process data. For example, they can add, subtract, or compare data to find patterns. **Turning data into information** - Data alone is like having puzzle pieces without knowing what they make. IT professionals use their programs to turn this data into useful information. It's like solving the puzzle and seeing the whole picture **Visualising information** - Sometimes, it's easier to understand information when it's shown in pictures or graphs. IT experts create charts and graphs to help people see and understand the data better. **Decision making** - Now that we have meaningful information, people (like business owners or doctors) can use it to make decisions. For example, a store owner might use data to decide what products to stock. **Security** - IT professionals also play a vital role in keeping data safe. They "build locks and walls" (security measures) to protect data from hackers and other threats **Problem solving** - When something goes wrong with data or information systems, IT professionals, including programmers, are like detectives. They figure out what's wrong and fix it. **Innovation** - IT experts are always inventing new ways to do things with data. They create new apps, software, and technology that make our lives easier and more fun.	Do with D.2, D.3 and D.4 Learners need to understand the role of information technology (IT) and IT professionals (programmers) in enabling transformation of data to meaningful information and decision making. IT professionals and programmers are like digital architects and builders. They collect, organize, and transform data into valuable information that helps people and businesses make decisions. They also make sure this data is safe and use technology to create exciting new things. So, when you see a cool app or a helpful website, remember that IT professionals and programmers are the ones behind the scenes making it all work. IT professionals, therefore, play a crucial role in transforming raw data into meaningful information through data processing. They use mathematics and statistics to analyse the information, make predictions, and guide decisions. Simulations allow them to test different scenarios and understand how various factors interact without real-world consequences. These aspects of IT contribute to better understanding, efficient decision-making, and advancements across various fields and industries.
D.2 Recognise that he or she is living as citizens in a digital world.	**Link to D.4 and D.6**
Revise the concepts of good password management. Revise the concept and dangers of sharing information like personal information, pins, usernames and passwords are recognised. Reference the use of passwords in class when logging and working in the current school environment. Understand that we leave a digital footprint when working with digital devices/computers/cell phones and present digital footprint concepts.	Revise and extend concepts of • privacy and security using good passwords • digital footprint
D.3 Demonstrate an understanding of the concept of a computing device.	**Link to D.1, D.2 and D.7 and C.2 and C.3**
Example activity – Computing devices Provide learners into small groups. Each group get one computing device to explore and report on: **Personal Computers (PCs):** Desktops, laptops, and workstations are common computing devices used for a wide range of tasks, from word processing and gaming to programming and data analysis	**Learners need to** • Understand that, generally, a computing device, in general, is an electronic machine or system designed to

Content (Grade 7 / Term 3)	Notes/Examples
Smartphones and Tablets: These mobile devices are powerful computing devices that allow users to make calls, send messages, browse the internet, run applications, and more **Servers:** Servers are specialized computing devices used to provide services, store data, and host websites or applications. They often operate 24/7 and are optimized for reliability and performance **Embedded Systems:** These are specialized computing devices designed for specific functions and often built into other products. Examples include the computer inside a car for engine control or the microcontrollers in household appliances **IoT Devices:** Internet of Things (IoT) devices, such as smart thermostats, security cameras, and wearable fitness trackers, have computing capabilities to collect data and perform tasks related to their specific functions **Gaming Consoles:** Devices like gaming consoles (e.g., Xbox, PlayStation) are specialized computing devices designed for playing video game **Routers and Modems:** These devices are used to manage network connections and enable internet access	process and manipulate data using a set of instructions or programs • Provide examples of computing devices and their main purpose
D.4 Identify the common uses of ICT in the real world	**Link to D.1**
Differentiate between the WWW (software related) and Internet (hardware related).  **Internet and WWW** **Internet** • Internet itself is a global, interconnected *network* of devices (computing devices, e.g. computer, network devices, e.g. router, etc.) • Supports a wide variety of interactions and communications between its devices https://youtu.be/x3c1ih2NJEg **WWW**  • WWW is one set of software services running on the Internet • Subset of these interactions and supports websites and URLs **Example activity:** Watch the videos https://youtu.be/J8hzJxb0rpc and https://youtu.be/r9Qw1iu3q6M Make notes while watching. Provide questions beforehand e.g. What is the difference between www and internet? Are webpages static or dynamic? What do webpages consist of? Use a KWLS chart and complete the columns.	Learners need to • Understand what the Internet is • Understand what the World Wide Web (WWW) is • Distinguish between the Internet and the WWW • Understand what is required to connect to the internet (refer to Term 2 D.5)
D.6 Explain how the adaptation of technology impacted the world we work and live in	**Link to D.1 – D.5 from previous terms**
We live in a connected world! - Introduction to the simplified concept of networks, e.g., network sharing and storage on the cloud. The cloud is like a virtual storage and processing space on the internet. It's where data can be stored and accessed from anywhere. **Example activity 1**:  Explain that cloud storage is a service that allows users to store and access their data (such as documents, photos, videos, and more) over the internet instead of on their physical devices (computers or smartphones). Discuss importance of cloud storage, such as backing up important files, sharing documents with others, and accessing files from any device with an internet connection. **Connecting Cloud Storage to Networks:** Review concept of networks, devices connected to one another using a network, including the internet and can be wired (cables) or wireless (Bluetooth, Wi-Fi, cellular). Draw a simple diagram showing a cloud symbol connected to various devices using lines representing network connections. Discuss how devices connect to the internet and access cloud storage services through network connections. **Cloud Storage services:** Explain that there are various cloud storage services available, such as Google Drive and OneDrive. Mention that each service may offer different features and storage capacities. Provide a brief overview of a popular cloud storage service.	Learners need to • understand that cloud storage (network of servers (big computers) connected to the internet) is a service that allows one to store one's files and photos online, instead of on one's device (on a hard disk). • understand that one can access these files from any device, anywhere, if one has an internet connection • provide an example of a cloud storage service • list the benefits and risk of cloud storage

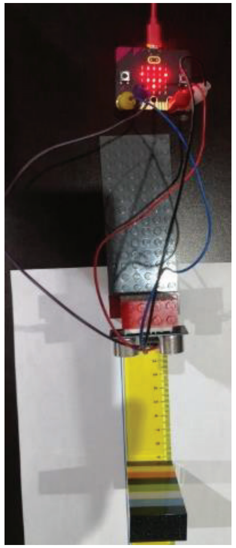
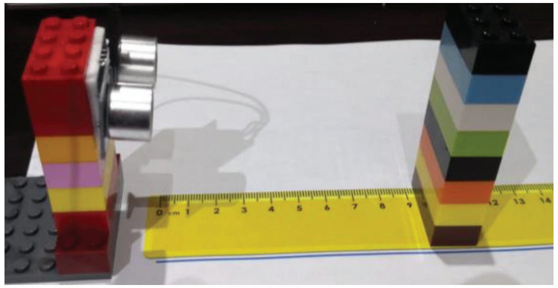
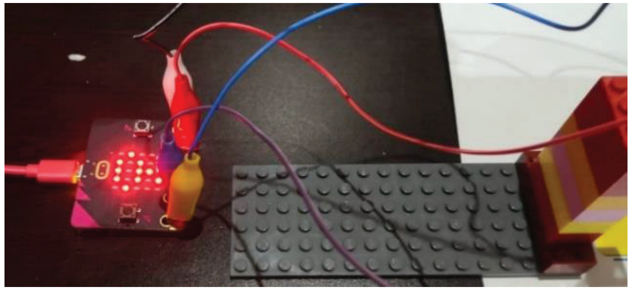
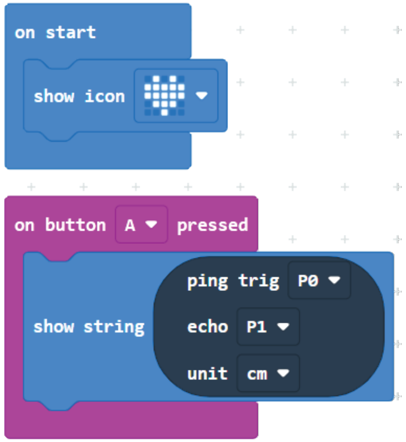
Content (Grade 7 / Term 3)	Notes/Examples
Benefits and Risks: Discuss the benefits of using cloud storage, such as data backup, easy sharing, remote access, and collaboration. Highlight potential risks, include data privacy and importance of strong passwords. **Hands-On Activity:** If possible, give learners access to devices (laptops or tablets) with internet access. Learners create a simple document or upload a picture to a cloud storage service like Google Drive. Guide them through the process **Discussion:** After the hands-on activity, ask learners about their experiences using cloud storage. What did they find easy or challenging? Did they notice any immediate benefits?	
D.7 Present a basic understanding of the concept of input processing and output.	**Link to C.1 – C.4**
Input, processing, output, communication (networks) The concepts of input, output, and processing take on specific meanings related to the flow of data and information within the network **Input**: In a network, input refers to the data or information that is received by the network from various sources such as other devices, users, sensors, or external data streams. Input data can take the form of text, files, multimedia, sensor readings, requests for resources or services, and more. Examples of input in a network include a user typing a web address into a browser (requesting a webpage), a sensor sending temperature readings to a server, or an email being received by a mail server. **Processing** Processing in a network involves the actions taken to handle and manipulate the input data. This can include routing data to its destination, filtering out unwanted or malicious data, performing calculations or transformations, and managing network resources. Network devices play a crucial role in processing data as they determine how data are transmitted Processing also includes tasks like data encryption for security, error correction. **Output** Output in a network refers to the data or information that is transmitted or sent out from the network to its intended recipients. This output can be directed to specific devices, users, Examples of output in a network include the delivery of a requested webpage to a user's device, sending an email to its recipient's inbox, or transmitting sensor data to a central monitoring system. Output may also include status updates, error messages, or acknowledgments to inform users or devices about the success or failure of data transmission.	Learners need to know that, in the context of a network Input-output-processing describes how data is received, processed, and transmitted to enable communication, resource sharing, and information exchange among devices and users connected to the network.
D.8 Interpret a pattern to represent or communicate a message or image	**Link to C.6 and C.7**
Continue by giving learners more examples of decimals to convert to binary and from binary to convert to decimal. **Example activity 1**: Use the concept of binary and determine the following: **Example activity 2**: Hold up one hand and let learners convert to binary. If a finger is up, it is a one, and if it is down, it is a zero. Ask learners what is the biggest number that can be represented if both hands and all 10 fingers are used. ✓✓✓✗✗ = (✗=0, ✓=1) ❍❑❑❍❍ = (❑=0, ❍=1) ☺☺☺☺☺ = (☺=0, ☹=1) ☒☒☒☒☒☒ = (☒☺=0, ⍓=1)	Revise binary code from Term 1
D.10 Demonstrate a basic proficiency in the application of digital skills.	**Link to D.7 and C.1 – C.7**
Strengthen file management proficiency (create folders, save files, give appropriate filenames, file organising). Learners use an application such as Paint to create sprites and backgrounds to be imported into the block-based apps.	File and Folder management Use paint

3.1.4 Term 4

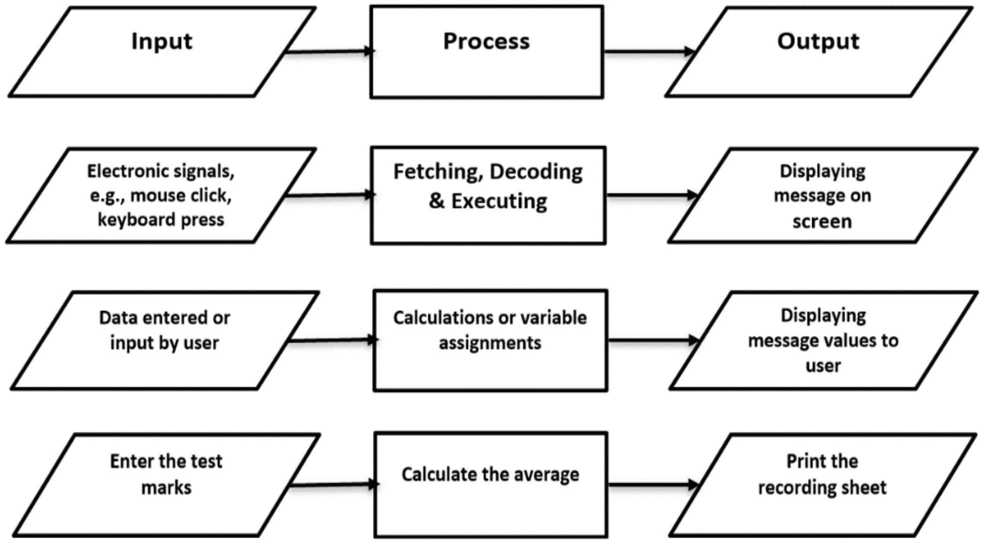
Content (Grade 7 / Term 4)	Notes/Examples
Coding	
C.1 Apply computational thinking (CT) skills to develop a set of logical instructions to solve a problem.	**Link to C.2 – C.7 and R.5 – R.7**
Example activity – Use computational thinking to develop an algorithm to solve a problem Divide learners into pairs. Each pair choose a game (e.g. a board game or a sport such as cricket, soccer, tennis, etc.). The pair must compile an algorithm using one sheet of paper to explain how the game works and how points are scored. Pairs explain their algorithms to other pairs who must evaluate if the algorithm explains the game and scoring correctly. Pairs explain how they used computational thinking to solve the problem.	Learners apply computational thinking to develop an algorithm
C.2 Present a simple coding solution using symbolic or written statements representing sequences of commands, single repetition, and conditional constructs.	**Link to C.1 – C.5**
Example activity – Display random list contents A list stores the names of 10 animals. When the green flag is clicked, the program picks a random animal from the list and displays the name of the animal as well as a picture of the animal (the costume number links to the. List number of the animal.    Animals 1 Kanga… 2 Ant 3 Giraffe 4 Elephant 5 Komo… 6 Shark 7 Ocelot + length 10 =  when ⚑ clicked repeat 10 set size to 100 % set Item to item pick random 1 to 10 of Animals switch costume to Item say Item for 2 seconds repeat 10 change size by 5 wait 0.2 seconds set Item to item Counter of Animals switch costume to Item say Item for 2 seconds Study the code and create a similar program, e.g. display treasures, etc.	**Note**: Common mistakes with lists: • Not initialising the list • Using the wrong index (counter) • Not checking for empty lists or not clearing lists when needed • Not using descriptive names • Not checking for out-of-bound errors (e.g. referring to list index 5 if the list only contains 4 items)
C.3 Interpret and execute a given symbolic or written set of commands	**Link to C.1, C.2, C.4 and C.5**
C.4 Debug a given symbolic or written set of instructions	**C.3, C.4 and C.5 done together** When learners write their own programs, the need to debug and evaluate the code for potential improvement. **Remember:** Effective problem decomposition is a key skill in programming Note As programs get bigger and more complex, it is important that learners learn plan a program. If code is not planned well before writing it, it can lead to difficulties when they need combining structures, as they may not have a clear roadmap for how different structures should work together.
C.5 Evaluate a given solution towards potential improvement	
Example activity – Game with two sprites **Sprite 1 (Hammer) code**  when this sprite clicked change TryCount by 1 turn ↺ 45 degrees wait 0.1 seconds turn ↻ 45 degrees if touching Sprite2 ? then change Counter by 1 broadcast splat when ⚑ clicked forever go to x: mouse x y: mouse **Sprite 2 (Troll) code**  when ⚑ clicked forever wait 1 seconds go to x: pick random 110 to -150 y: pick random 110 to -150 when I receive splat switch costume to troll2 wait 0.2 seconds switch costume to troll1 Implement the code on the left in a block-based coding app. Explain what the code does. Now, write a similar game of your choice.	

Content (Grade 7 / Term 4)	Notes/Examples
C.6 Recognise and interpret patterns in symbolic sets of data or visualisations.	
Example activity - Secret Digits Beavers use symbols instead of house numbers. They use the table on the right to translate the symbols to numbers Example Five is written as :  What is the house number of this beaver house? 2021-TS-Junior-Question-Paper.pdf (olympiad.org.za)  	
C.7 Create or complete a pattern to represent a data set	
Example activity – Tree Sudoku Beavers planted 9 trees in a 3 x 3 field. The trees had three different heights. No line (horizontal or vertical) had two trees of the same height. The beavers observed how many trees they could see from every position where there was a sign. They then wrote this number on the sign to indicate how many trees they can see in the line (horizontally or vertically from that point. However, when looking along a line of trees, the beavers could not see trees that were planted behind taller trees! Unfortunately, the rain has washed some of the numbers off some signs Short , Medium , and Tall   signs for vertical lines and  signs for horizontal lines Use the remaining numbers on the signs as an indication to show them which trees were planted where. Write down the height of trees (S (small), M (medium), T (tall) in the in the block on the grid.   2021-TS-Junior-Question-Paper.pdf (olympiad.org.za)	**Note:** Concrete activities remain important as literature suggests that the primary weakness of today's pedagogy of programming is that it does not provide enough opportunity for the novice to develop concrete operational skills, via the correct types of exercises…due to too much emphasis on writing large amounts of code, and problem solving. By identifying patterns, we can predict what will come next and what will happen again and again in the same way. In Computer Science/coding we analyse patterns in data and make predictions and generalisations based on the pattern analysis.

<table>
<tr><th>Content (Grade 7 / Term 4)</th><th>Notes/Examples</th></tr>
<tr><td>Robotics</td><td></td></tr>
<tr><td>R.3 Outline the different components of a robot</td><td></td></tr>
<tr><td>Learners need to provide a basic definition and purpose of the following basic electronic components of a robot:
<table>
<tr><th>Component</th><th>Basic definition</th><th>Purpose</th><th>Practical application example – May be used to:</th></tr>
<tr><td>Ultrasonic Sensor:
</td><td>A sensor that emits and receives ultrasonic waves to measure distances.</td><td>Used for obstacle detection, navigation, and proximity sensing in robotics.</td><td>Provides distance measurements for obstacle avoidance, navigation, and object detection.</td></tr>
</table></td><td>Learners must describe the ultrasonic sensor in term of
• basic definition and
• purpose:</td></tr>
<tr><td>R.5 Design a simple artefact based on a set of design instructions</td><td>Done with R.6</td></tr>
<tr><td>R.6 Mimic the operations of a robot</td><td rowspan="2">R.5 and R.6 done together

Note:
It is important that learners plan their projects.
Planning should include identifying resources required as well as planning the code.
If a project is not planned well before building it and writing code to realise the artefact, it can lead to difficulties when they need combining components and structures, as they may not have a clear roadmap for how different components and structures should work together.</td></tr>
<tr><td>Example Project – Ultrasonic Sensor Distance Example with microcontroller




on start
show icon
on button A pressed
show string ping trig P0 echo P1 unit cm</td></tr>
<tr><td>R.7 Create, test and execute a set of robotic instructions</td><td>Link to R.6</td></tr>
<tr><td>Example activity 1 - Timer and score keeper for thirty-seconds game.
Create a microcontroller program that can act as a timer for the game of 30-seconds and to keep score for two teams.
The game is played with two groups, group A and group B.
Shake the microcontroller to start counting down, Display the seconds starting from 30 and counting down.
“Draw” a card from a bowl/box with a question to answer.
When the timer reaches 0, make a sound.
If group A gets an answer right, press Button A to keep the score (add 1 point) and display their score.
If group B gets an answer right, press Button B to keep the score (add 1 point) and display their score.
Press both buttons A and B together to determine which group has won and display a message to say which group is the winner.</td><td>Learners create an alarm with PIR or Ultrasonic sensor and additional component e.g., External buzzer, LED.</td></tr>
</table>

Content (Grade 7 / Term 4)	Notes/Examples
Digital Concepts	
D.1 Outline the concept of technology and purpose of information technology (IT)	**Link to D.3, D.4 and D.7**
Case Study: The Stolen Artefacts – Solving a mystery using IPO tables **Instructions:** 1. **Introduction**: Revise the concept of input, processing, and output. Explain that these steps are crucial for solving problems effectively. 2. **The Mystery**: Present the mystery scenario to the students: "The Stolen Artefacts." Explain that during a school trip to a local museum, several valuable historical artefacts have gone missing. Present the problem to the learners and mention that they will play the role of detectives to solve this mystery. 3. **Input Phase**: Discuss with the learners what information or clues they need to gather to solve the mystery. Write these on the whiteboard. For example, they might need information about when the artefacts were last seen, who had access to the museum, and if there were any security breaches. Encourage learners to think critically and ask questions about what information is essential to solve the case. 4. **Processing Phase**: Discuss how they will process the information they've gathered. Encourage them to brainstorm ideas and analyse the data they have collected. Guide the learners to think about possible scenarios and motives for the theft. They should consider who might have stolen the artefacts and why. 5. **Output Phase**: Discuss how they will present their findings or conclusions to solve the mystery. This might include creating a detective report, sharing their theories, or identifying the possible thief and motive. Encourage learners to communicate their theories and reasoning effectively. 6. **Mystery-Solving Time**: Divide the learners into small groups and provide them with the mystery scenario and any clues (if you choose to use envelopes or containers). In their groups, learners should follow the input, processing, and output steps to solve the mystery by creating an IPO table. They can record their findings and reasoning on a sheet of paper. 7. **Conclusion**: Discuss how input, processing, and output were used in solving the mystery. Ask learners to reflect on what they learned about problem-solving through this activity.	Learners apply the concepts of input, processing, and output to solve the mystery of stolen historical artefacts.
D.2 Recognise that he or she is living as citizens in a digital world.	**Link to D.6**
Case Study Scenarios: **Scenario 1: Social Media Dilemma** Mpho discovers that her best friend, Sam, has been posting hurtful and misleading information about her on a social media platform. Mpho is torn between confronting Sam publicly or privately. What should she do, and what are the potential consequences of her decision? **Scenario 2: Online Bullying** Alex witnesses a classmate being cyberbullied on a chat platform. Should Alex intervene, report the incident, or ignore it? What ethical considerations come into play when deciding how to respond? **Scenario 3: Privacy vs. Convenience** Emily is tempted to download a popular app that requests access to her personal information and location. She finds the app incredibly useful but worries about her privacy. How can Emily balance convenience with protecting her personal data? **Scenario 4: Digital Addiction** Chris spends hours every day playing video games and scrolling through social media, neglecting his responsibilities and health. How can Chris recognise and address his digital addiction, and what strategies can he implement to regain control of his life? **Answer the following questions:** **Scenario 1: Social Media Dilemma** 1. What are the immediate emotions and reactions that Mpho might be experiencing upon discovering Sam's hurtful posts? 2. What are the potential consequences of Mpho confronting Sam's publicly on social media? How might this impact their friendship and others who see the exchange? 3. What are the potential consequences of Mpho addressing the issue privately with Sam? How might this approach affect their relationship and resolution of the problem? 4. What ethical principles should guide Mpho in deciding whether to confront Sam publicly or privately? **Scenario 2: Online Bullying** 1. What responsibilities do individuals like Alex have when witnessing cyberbullying on an online platform? 2. What are the potential consequences of Alex intervening and acting against the cyberbully? 3. What are the potential consequences of Alex reporting the cyberbullying incident to the platform administrators? 4. What ethical considerations should Alex consider when deciding how to respond to the situation? **Scenario 3: Privacy vs. Convenience** 1. What are the benefits and drawbacks of Emily's decision to download the app that requests access to her personal information and location? 2. What privacy risks does Emily face if she proceeds with downloading the app?	

Content (Grade 7 / Term 4)	Notes/Examples
3. How can Emily balance her desire for convenience with protecting her personal data and privacy? 4. What steps can Emily take to make an informed decision about using the app? **Scenario 4: Digital Addiction** 1. What signs of digital addiction are evident in Chris's behaviour? 2. How might Chris's digital addiction impact his responsibilities, relationships, and overall well-being? 3. What strategies can Chris implement to recognize and address his digital addiction? 4. How can Chris regain control of his life and find a healthier balance between digital activities and other aspects of his life?	
D.3 Demonstrate an understanding of the concept of a computing device.	**Link to D.1**
D.4 Identify the common uses of ICT in the real world	**Link to D.5**
Education, Entertainment chatbots	
D.5 Differentiate between the components of an ICT system	**Link to D.4**
D.6 Explain how the adaptation of technology impacted the world we work and live in	**Link to all D.1 to D.5, D.7 and D.10**
Example activity: Let learners answer the following questions: 1. How has technology changed the way people work and live over the past few decades? 2. Indicate how technology has affected the types of jobs people have and how they do those jobs? 3. How has technology made it easier for people to communicate and connect with others around the world? 4. How do you think technology has influenced the things we watch, play, and enjoy for entertainment? 5. Indicate how technology has improved our access to information and learning? 6. Can you think of examples where technology has helped improve our health or the way we receive medical care? 7. How has technology changed the way people interact socially and the kinds of activities they do together? 8. What concerns do people have about privacy and safety in our digital world, and how can we address them? 9. Do you think technology has made our lives better overall? How? **10.** What new technologies do you think will be important in the future, and how might they change our lives?	
D.7 Present a basic understanding of the concept of input processing and output.	**Link to D.7 and C.2 – C.5**
Explain different types of input, processing and output **Example activity**: Create an IPO table to calculate the area of a rectangle. The user will provide the length and width of the rectangle as inputs, and the program will process this information to output the calculated area. Execution can be applied in a coding activity (Link to D.10). 	Link to robotics (e.g. sensors, etc.) and robotics apps Link to coding apps and problems.

<table>
<tr><th>Content (Grade 7 / Term 4)</th><th>Notes/Examples</th></tr>
<tr><td>D.8 Interpret a pattern to represent or communicate a message or image</td><td>Link to D.9, D.10 and C.1</td></tr>
<tr><td>Example activity – "Binary Bingo"
Objective: Reinforce understanding of binary through a competitive Bingo game.
Materials Needed: Bingo cards with binary numbers. Binary Bingo Card (bingobaker.com), E.g.
Instructions:
Create Bingo cards with binary numbers, prepare a list of corresponding decimal numbers, and provide each learner with a card.
Revise converting from decimal to binary.
Call out decimal numbers, learners cover corresponding binary numbers on their cards.
After a Bingo, review binary numbers called and their decimal equivalents.
Offer prizes to Bingo winners.
Optional: Play more rounds with different Bingo patterns. / Reverse the came by supplying cards with decimal numbers and write binary number on the board
Summarise key points about binary and its role in computers.

<table>
<tr><th>B</th><th>I</th><th>N</th><th>G</th><th>O</th></tr>
<tr><td>0001111</td><td>0010111</td><td>0011111</td><td>0110010</td><td>0111101</td></tr>
<tr><td>0001000</td><td>0011011</td><td>0101000</td><td>0110001</td><td>1001001</td></tr>
<tr><td>0000111</td><td>0010011</td><td>Free!</td><td>0111100</td><td>1000000</td></tr>
<tr><td>0000110</td><td>0011010</td><td>0100000</td><td>0101110</td><td>0111110</td></tr>
<tr><td>0000011</td><td>0011100</td><td>0100011</td><td>0110100</td><td>1000100</td></tr>
</table>

<table>
<tr><th>B</th><th>I</th><th>N</th><th>G</th><th>O</th></tr>
<tr><td>0001011</td><td>0010001</td><td>0100010</td><td>0101111</td><td>1000000</td></tr>
<tr><td>0001100</td><td>0010110</td><td>0101001</td><td>0111000</td><td>1000011</td></tr>
<tr><td>0001010</td><td>0011101</td><td>Free!</td><td>0110010</td><td>1000100</td></tr>
<tr><td>0001111</td><td>0010101</td><td>0101101</td><td>0111001</td><td>0111101</td></tr>
<tr><td>0000110</td><td>0011001</td><td>0101100</td><td>0110001</td><td>0111110</td></tr>
</table>
</td><td>Binary number system – revise and extend</td></tr>
<tr><td>D.9 Create a pattern to represent or communicate a message or image</td><td>Link to D.8 and C.1</td></tr>
<tr><td>Link to D.8. and D.10. Learners create their own Bingo cards and corresponding decimal numbers.</td><td>Do with D.8</td></tr>
<tr><td>D.10 Demonstrate a basic proficiency in the application of digital skills.</td><td>Link D.8 and D.9.</td></tr>
<tr><td>Link to D.8. and let learners create their own Binary Bingo cards in Paint.</td><td>Do with D,8 and Cs</td></tr>
</table>

3.2 GRADE 8

Note:
Teachers must include the following competencies and content in their Annual Teaching Plan (ATP), distributed across the terms and sequenced, organised and grouped in a manner that will facilitate learning in a manner that will make sense for learning and teaching, maximize the learners' learning outcomes and achievement. and in a way that will make optimal use of time and resources. Some competencies could also be combined in bigger/more complex activities/programs.

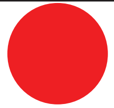
3.2.1 Term 1

Content (Grade 8 / Term 1)	Notes/Examples
Coding	
C.1 Apply computational thinking (CT) skills to develop a set of logical instructions to solve a problem.	**Link to C.1 – C.7, R.5 – R.7**

Example activity

In Grade 7, you have looked at different geometrical shapes: (square, pentagon, hexagon, heptagon and octagon) and the pattern/ relationship between these.

You have noticed the following/identified the following

	Square	Pentagon	Hexagon	Heptagon	Octagon	Circle
Example activity Shape						
Number of sides	4	5	6	7	8	(360)
Number of angles	4	5	6	7	8	(360)
Size of the interior angle	90°	72°	60°	51,4°	45°	1°
Sum of interior angles	360°	360°	360°	360°	360°	360°

patterns/trends/relationships:

- As the number of sides **increases**
  - the size of the interior angles **decreases**
  - the form of the shape gets closer to a circle
- The **sum** of the interior angles of a shape equals **360°** and a **circle** is also 360°

This information helped you to make more observations:

- The size of the interior angles for each shape can be calculated by dividing 360° by the number of sides, e.g. for the pentagon the interior angles can be calculated as 360/5 = 72.

There is a relationship between the of sides, the size of the interior angles and a circle (360°).

- These patterns and relationships help to **generalise** the solution/algorithm for drawing shapes by finding general principles that create patterns. Look at the following:

Square (4 sides)	**Pentagon** (5 sides)	**Circle** (**360** '*sides*')	**N** sides
Repeat **4**x Turn 360/**4** (**90°**) Move 360/**4** steps	Repeat **5** times Turn 360/**5** degrees Move 360/**5** steps	Repeat **360** times Turn 360/**360** degrees Move 360/**360** steps	Repeat **n** *times* Turn 360/**n** degrees Move 360/**n** steps

The **generalised algorithm** for drawing shapes allows us to draw any shape – from a triangle to a circle! – by understanding the relationship between the steps; the size of the angles and a circle (a circle as 360 "sides" that forms the round shape).

Notes/Examples:

Done with C.2 and C.5, C.6 and C.7

Show learners an octagon within a circle to see that a shape gets closer to resemble a circle as the number of sides increases and the size of the interior angles decreases

Octagon within a circle

Learners use computational thinking (and specifically pattern recognition) to reason about a generalised formula for drawing a shape.

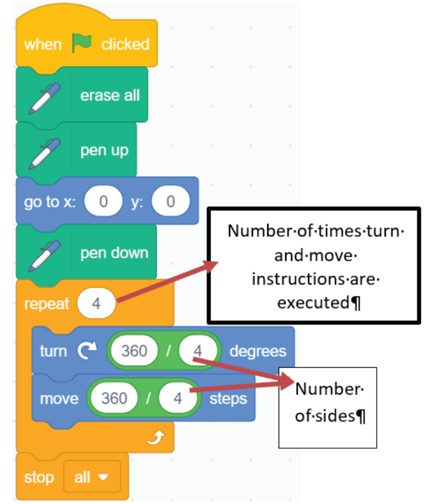
Recognising patterns helps you to identify similarities, shared characteristics and/or differences within or between problems. This helps to make **predictions** or to **generalise** solutions. If one can identify a pattern, one is generally able to predict, for example, the next item in a series or the next step in completing a task.

How are patterns and pattern recognition helpful?

- Identifying patterns can point you to an existing solution that you can use to solve a new problem.
- Patterns help us to make predictions
- Pattern recognition helps us to generalise

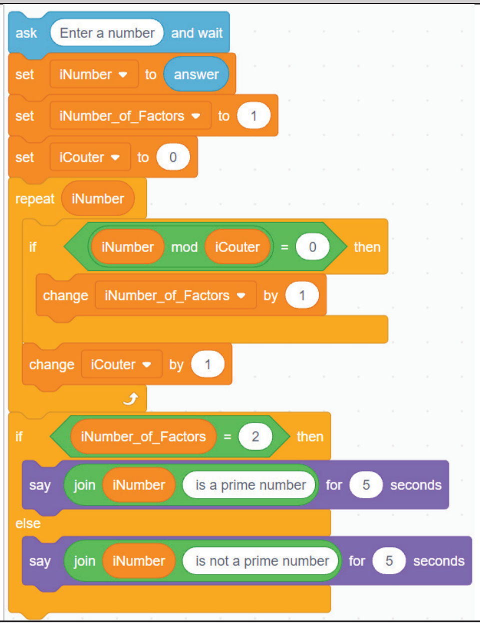
To **generalise** is to look at specific cases; identify a pattern or relationship that will always be true; then represent the pattern in symbolic format.

Content (Grade 8 / Term 1)	Notes/Examples
C.2 Present a simple coding solution using symbolic or written statements representing sequences of commands, single repetition, and conditional constructs.	**Link to C1, C.3 – C.7 and R.5 – R.7**

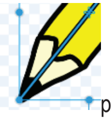
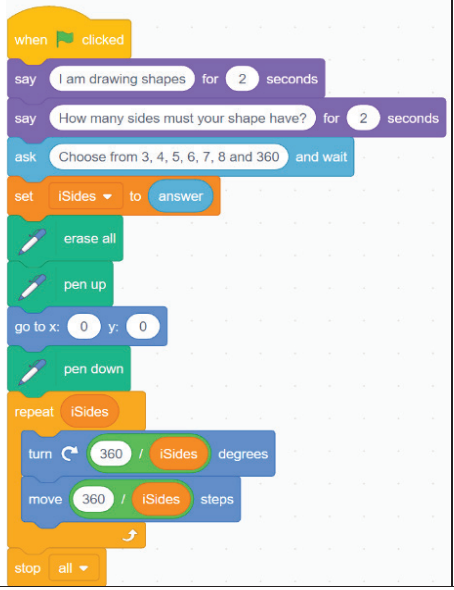
Content (Grade 8 / Term 1)	Notes/Examples
Example activity 1 Provide learners with the code on the right. Learners need to study the code and follow the instructions below: 1. Run the code 2. Change the number of sides and steps (all the 4's) to 5. See what happens. 3. Now change the number of sides and steps to 6 and 8 respectively and watch what happens. 4. Finally, change the number of sides and steps to 360 and watch what happens.  Number·of·times·turn·and·move·instructions·are·executed¶ Number·of·sides¶	The purpose of activity 1 is for learners to see how the shape changes when the number of sides is changed withing the 'formula'. Ensure that learners understand that by only changing the number of sides, we can use the code to draw different shapes.
C.3 Interpret and execute a given symbolic or written set of commands	**Link to C.1, C4, and C.6**
Example activity – Introduce: When I start as a clone The following game lets balloons burst when one clicks on them and use a timer to show the time it takes you to burst the balloons. Run the code and explain how the program works. Now write a similar game, 	**Note:** Literature suggests that it is important that learners must also read and explain in plain language (their own words) what the code does. This type of activities should be done unplugged (pen-and-paper) and only implemented after learners explained the results. Many of these types of exercises are necessary to ground concepts, skills and understanding of algorithms and coding. **Note:** While learners should be able to describe what each line (block) of code does, (describing a code segment line-by-line/block-by-block) it is very important that learners explain the overall purpose of the code (program), i.e. what the program does/the purpose of the program is (not a line-by-line description).
C.4 Debug a given symbolic or written set of instructions	**C.3 and C.4 done together and C.6**

```
when 🏴 clicked
erase all
pen up
go to x: 0 y: 0
pen down
repeat 4
    turn ↻ 360 / 4 degrees
    move 360 / 4 steps
stop all
```

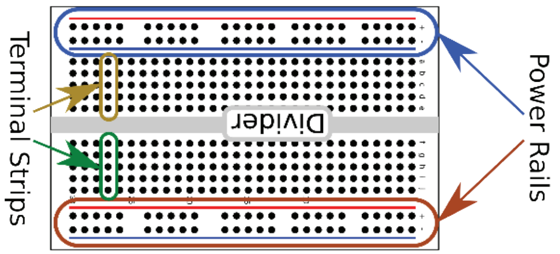
```
set BestTime to 9999

when 🏴 clicked
reset timer
set BurstCount to 0
repeat 10
    glide 0.1 secs to random position
    next costume
    set size to pick random 30 to 100 %
    create clone of myself
hide
wait until BurstCount = 10
if timer < BestTime then
    set BestTime to timer

when I start as a clone
show
forever
    glide 5 secs to random position

when this sprite clicked
change BurstCount by 1
start sound Pop
delete this clone
```

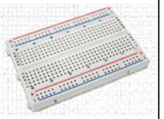
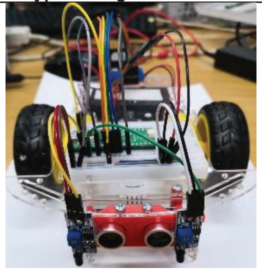
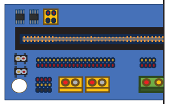
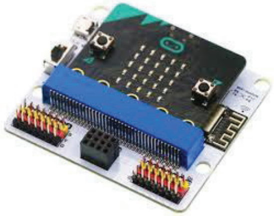
Content (Grade 8 / Term 1)	Notes/Examples
Example activity - Determine if a number is a prime number A prime number has only two factors: Then number itself and 1. The following program was written to determine if a number is prime or not, However, it does not work correctly. Use a trace table to find the error and correct the program 	**Note:** Initialisation is the assignment of an initial value to a variable and is an important concept in coding as it allows the program to start with predefined values and avoid undefined behaviour of programs such as garbage in and garbage out (GIGO). Generally, mistakes with initialisation are common.

```
ask Enter a number and wait
set iNumber to answer
set iNumber_of_Factors to 1
set iCouter to 0
repeat iNumber
    if iNumber mod iCouter = 0 then
        change iNumber_of_Factors by 1
    change iCouter by 1
if iNumber_of_Factors = 2 then
    say join iNumber is a prime number for 5 seconds
else
    say join iNumber is not a prime number for 5 seconds
```

C.5 Evaluate a given solution towards potential improvement	
Example activity - Improve code Provide learners with the code from C.2 Request learners to improve the code using a variable so that the program could ask the user how many sides their shape must have (giving them the following options 3, 4, 5, 6, 7, 8 or 360) **Note:** Calibrate the pen (sprite) to ensure precise drawing. Go to costumes and ensure that  pen point is aligned with the bottom left dot. **Possible solution** 	Previously, learners drew a square using fixed values. Learners will now be able to develop code that uses the generalised formula, using a variable, to draw any shape the user wishes to draw. (Refer to C.2) To **generalise** is to look at specific cases; identify a pattern or relationship that will always be true; then represent the pattern in symbolic format. The algorithm in C.2 will now be improved using a variable to be able to draw the shape a user requests (using input from the user)

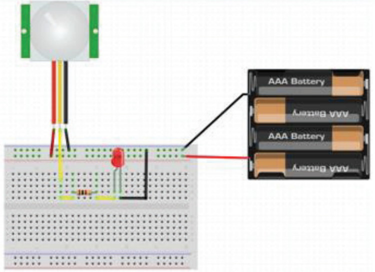
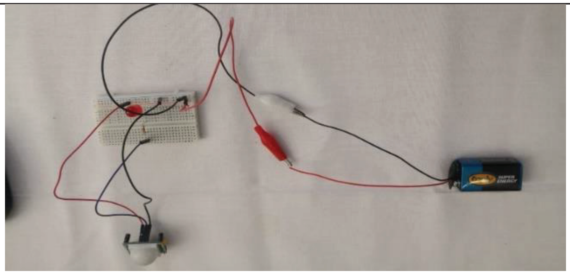
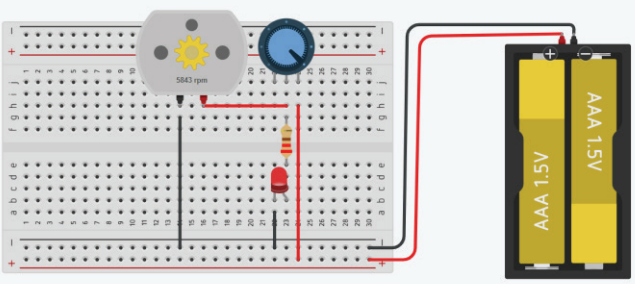
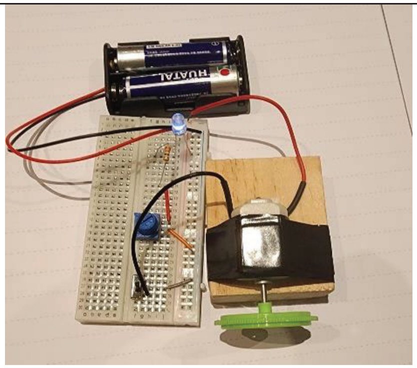
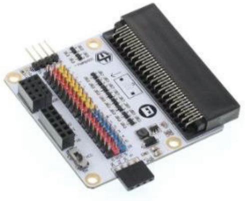
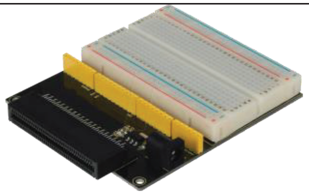
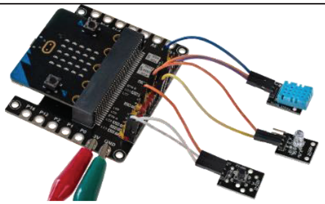
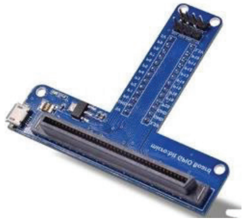
```
when clicked
say I am drawing shapes for 2 seconds
say How many sides must your shape have? for 2 seconds
ask Choose from 3, 4, 5, 6, 7, 8 and 360 and wait
set iSides to answer
erase all
pen up
go to x: 0 y: 0
pen down
repeat iSides
    turn 360 / iSides degrees
    move 360 / iSides steps
stop all
```

Content (Grade 8 / Term 1)	Notes/Examples
C.6 Recognise and interpret patterns in symbolic sets of data or visualisations.	**Link to C.1 – C.4 and R.5 and R.6 and D.6 and D.7**
Refer to C.1 – interpret and describe patterns for geometric figures and make conclusions based on the pattern provided	**Done with C.1 C.2 and C.5**
C.7 Create or complete a pattern to represent a data set	**Link to C.1 – C.4 and R.5 and R.6 and D.6 and D.7**
Refer to C.1 – complete pattern for geometric figures and abstract a generalised algorithm by identifying the general principles that create the pattern.	**Done with C.1 C.2 and C.5**
Robotics	
R.1 Explain what a robot is in simple terms.	**Link to R.2**
R.2 Identify different types of robots.	**Link to R.1**
Revise concept of a robot Learners need to investigate a robot in the security/ protection industry **Example activity- A robot/drone in the medical or agricultural industry** Learners investigate a robot/drone used in medical or agricultural industry. They need to outline • The main components of the selected robot and • Describe how the robot is used to achieve its goals. • Create a diagrammatical outline of the robot and its various components and extension capabilities. • Present a short report on how robots that are used in the security and protection industry could be used on public transport or on farms	**R.1 and R.2 done together** Learners can outline different types of robots in terms of their use and application, i.e., • Medical Robots • Agricultural Robots Virtual Customer Service Agents (in relation to automation and virtual robots used in the information systems space)
R.3 Outline the different components of a robot	**Link to R.5**
Example activity – Introduce learners to a breadboard as a prototyping board/tool  Source www.pngegg.com Note: A breadboard consists of a rectangular or square-shaped plastic base with multiple rows and columns of interconnected metal terminals. These terminals are organized in a grid pattern and are typically made of metal strips or spring clips. The key components of a breadboard include: **Base:** The plastic base provides structural support for the breadboard and houses the grid of electrical terminals. It's often color-coded, with white or beige being common colors. **Rows:** Rows of metal terminals run horizontally along the breadboard. Each row typically contains five interconnected terminals. These rows are labeled with numbers or letters for easy reference. Columns: Columns of metal terminals run vertically along the breadboard. Each column typically contains multiple interconnected terminals. These columns are often used for connecting components like integrated circuits (ICs) or sensors. **Power Rails**: Many breadboards have two long rows of terminals on the sides, often labeled as "+ (positive)" and "- (negative)." These are the power rails and are used for providing voltage and ground connections to the circuit. **Terminal Holes**: Small holes in the plastic base accommodate the insertion of component leads or wires. These holes connect to the underlying metal terminals. **Examples**:	Review the following electronic components in terms of providing a: • Basic definition • Purpose, and • how it could be used in simple physical computing and robotics projects. Learners must Build and prototype elementary circuits based on a GIVEN breadboard design (NOT circuit diagram) using simple components. **Introduce the learners to the concept of a breakout board / expansion board.** Outline the purpose of an expansion board and how it is used in robotics and physical computing projects. Outline the concept of GPIO pins and connectors. The learners should be able to differentiate between the following types of pins: • Voltage supply • GND • Digital pins • Analogue pins

Content (Grade 8 / Term 1)					Notes/Examples
Component	**Basic definition**	**Purpose**	**Practical application example – May be used to:**	**Example: Line following robot prototyped using a breadboard**	**NOTE:** For **all projects and experiments** in grade 8, the **learners must be instructed/guided** in terms of which pins to use for the applicable projects.
Breadboard 	A prototyping board with a grid of interconnected electrical terminals, used for building and testing electronic circuits without soldering.	Facilitates the quick and temporary assembly of electronic circuits, making it easier to prototype and experiment with various component connections.	Breadboards are valuable in robotics projects for creating and testing circuitry configurations, allowing for rapid iteration and customization of the electrical systems within a robot. They enable the connection of various components, sensors, and microcontrollers without the need for permanent soldering, making them ideal for the development and debugging phases of robot construction. Components can be inserted into the terminal holes, and the interconnected rows and columns simplify the process of creating complex circuit connections for prototyping and experimentation.		
Expansion board 	An expansion board, also known as a shield or daughterboard, is a separate circuit board designed to attach to a primary electronic platform, such as a microcontroller or a breadboard, to provide additional functionality, components, or connectivity options for specific applications.	The primary purpose of an expansion board is to extend the capabilities of the main platform, adding specialized features or components to enhance its functionality.	In robotics, expansion boards are commonly used to simplify the connection of specific components or subsystems to the main robot control system. For example, a robotics expansion board might include motor drivers, sensors, communication interfaces, or specialized processors tailored for robotic applications. These expansion boards streamline the integration of essential hardware components, making it easier for engineers and enthusiasts to build, customize, and control robotic systems efficiently.	 **Example: Microcontroller connected to an expansion board**	

Components for use in elementary projects include all previous components (excluding sensors and servos (for breadboard prototyping)) PLUS

Component	**Basic definition**	**Purpose**	**Practical application example – May be used to:**
DC Motors 	Electric motors that operate on direct current (DC) power.	Provide motion to various parts of a robot, such as wheels, arms, or grippers.	Powers robot movement, including wheel propulsion, arm articulation, and conveyor systems.
Polarized Capacitor 	An electrical component that stores and releases electrical energy.	Used to filter, stabilize, or store electrical energy in various robot circuits.	Stabilizes voltage levels, reduces noise, and stores energy in various robot circuits.
Potentiometer 	A potentiometer is a three-terminal variable resistor that allows manual adjustment of its resistance, typically by rotating a knob or slider. It produces a varying voltage output between its terminals as the knob is turned, which is proportional to the physical position of the knob.	Potentiometers are used to control and vary electrical resistance in a circuit. They enable users to adjust and fine-tune parameters like volume, brightness, speed, or position in various electronic devices.	In robotics, potentiometers find application in control systems, especially for mechanisms where precise adjustments are required. For instance, potentiometers can be used in robotic arms to determine the angle or position of a joint, allowing the robot to accurately control the movement of its limbs. Potentiometers can also be employed in user interfaces, enabling manual input for tasks like robot arm positioning, gripper control, or setting operating parameters.
External DC Power Source (e.g., Batteries) 	An independent source of electrical power, typically in the form of batteries.	Provides the primary source of energy to drive all robot components and systems.	Provides the necessary energy to operate the robot's components and systems.

Content (Grade 8 / Term 1)	Notes/Examples
Example 1: Simple electronic circuit to show the functioning of a PiR sensor.	
 	
Example 2: Simple electronic circuit built from given breadboard circuit diagram. The speed of the DC motor and brightness of the LED is regulated with the potentiometer.	
 Breadboard design (given to learners)  Equivalent build by learner (In this example the learner decided to add a switch)	
 Sample Expansion board  Expansion board with breadboard  Example of expansion board with sensors connected.  T-Type expansion board	
Also see: https://tinkercademy.com/tutorials/microbit-breakout-board/	

Content (Grade 8 / Term 1) | **Notes/Examples**

Notes:
Breakout/Expansion boards are hardware components that enhance the capabilities of a microcontroller or single-board computer like the micro:bit. They are widely used in robotics and physical computing projects to simplify hardware connections, extend functionality, and enable more complex projects.

The main purposes of an expansion board for microcontrollers are:

- ***Hardware compatibility:*** Breakout boards make it easy to connect various hardware components such as sensors, motors, displays, and more to the microcontrollers.
- ***Ease of use:*** Breakout boards simplify the process of connecting and disconnecting hardware components.
- ***Protection:*** Many expansion boards provide protection for the microcontrollers by offering voltage regulation and current limiting.
- ***Signal conditioning:*** Some breakout boards include signal conditioning circuitry that can filter or amplify sensor inputs.
- ***I/O expansion:*** microcontrollers have a limited number of I/O pins. Expansion boards often provide additional GPIO pins, allowing you to connect more components simultaneously.
- ***Specialized functions:*** Expansion boards can include specialized functions like motor drivers, audio amplifiers, or communication interfaces to extend the microcontrollers capabilities for specific applications.

Breakout boards are particularly valuable in robotics and physical computing projects for the following reasons:

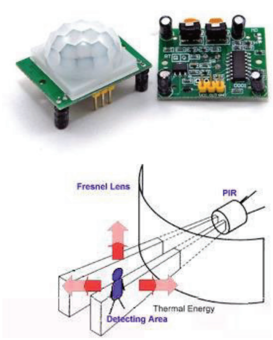
- ***Sensor integration:*** Robotics projects often require a wide array of sensors. Expansion boards simplify the connection and usage of these sensors.
- ***Motor control:*** Many robots require motor control to move and perform tasks. Expansion boards with motor drivers can control various types of motors.
- ***Display and feedback:*** Expansion boards may include LED displays or small screens that can convey information or feedback to users.
- ***Communication:*** Some expansion boards come equipped with communication interfaces to enable remote control and data exchange between the microcontrollers and other devices.
- ***Audio output:*** In robotics and interactive projects, audio feedback or speech synthesis is sometimes needed. Expansion boards can include audio amplifiers and speakers for this purpose.

R.5 Design a simple artefact based on a set of design instructions

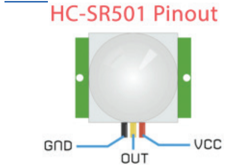
R.6

Ultrasonic and PIR functional operation

PIR Sensor



Source: https://osoyoo.com/2019/01/23/graphical-programming-tutorial-for-arduino-pir-motion-sensor/



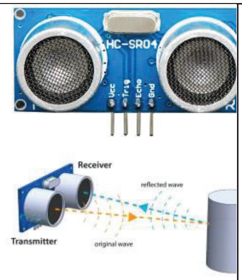
Source: https://datasheethub.com/hc-sr501-pir-motion-sensor-module/

PIR sensors work based on detecting changes in infrared radiation emitted by objects.
They contain a passive sensor that detects heat (infrared radiation) emitted by objects in their field of view.
When a warm object moves across the sensor's detection zone, it generates a change in the detected infrared radiation, triggering the sensor.

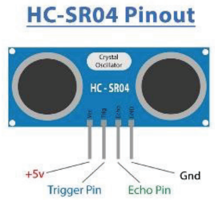
Typical pins on a PIR

- VCC (Power Supply): This pin is used to connect the PIR sensor to the power supply. It typically operates at a voltage of 3.3V or 5V, depending on the sensor's specifications.
- OUT (Output): The OUT pin is where the sensor sends its output signal. When motion is detected, this pin typically goes high (logic level HIGH) or low (logic level LOW), depending on the sensor's configuration.
- GND (Ground): This pin is connected to the ground or 0V reference of the power supply.

Ultrasonic sensor



Source: https://howtomechatronics.com/tutorials/Arduino/ultrasonic-sensor-hc-sr04/



Source: https://www.circuits-diy.com/hc-sr04-ultrasonic-distance-sensor/

Ultrasonic sensors use sound waves (ultrasonic pulses) to detect the presence of objects.
These sensors emit high-frequency sound waves and measure the time it takes for the waves to bounce back after hitting an object.
By calculating the time taken for the sound waves to return, the sensor can determine the distance to the object and detect motion based on changes in this distance.

Typical pins on an Ultrasonic Sensor

- VCC (Power Supply): Like PIR sensors, the VCC pin is used to connect the ultrasonic sensor to the power supply. It typically operates at a voltage of 3.3V or 5V.
- Trigger (or TRIG): The Trigger pin is used to initiate the ultrasonic measurement. When you send a brief high-level pulse (typically 10 microseconds) to this pin, the sensor emits an ultrasonic sound pulse.
- Echo: The Echo pin is where the sensor sends a signal back after it has received the ultrasonic pulse and detected the reflection from an object. The duration of the pulse on this pin is proportional to the distance to the object.
- GND (Ground): The GND pin is connected to the ground or 0V reference of the power supply.

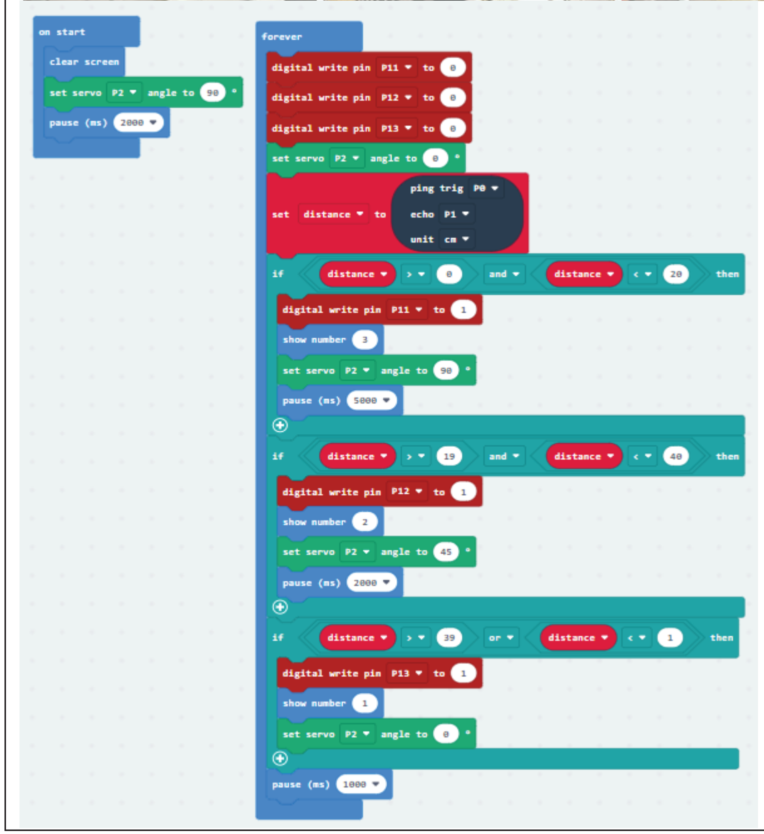
Link to R.5 and R.6 are done together

The learners need to be able to outline the difference in the functioning and operation of a PIR sensor as opposed to an ultrasonic sensor.
The learners also need to understand how the various pins on each of the sensors are used.

Content (Grade 8 / Term 1)

Example 1 – Automatic dustbin

In the following example an expansion board is used which allows one to connect more components to the microcontroller. A dustbin lid is controlled to open and close based on the movement of a servo motor, which is triggered by an ultrasonic sensor.



```
on start
    clear screen
    set servo P2 ▾ angle to 90 °
    pause (ms) 2000 ▾

forever
    digital write pin P11 ▾ to 0
    digital write pin P12 ▾ to 0
    digital write pin P13 ▾ to 0
    set servo P2 ▾ angle to 0 °
    set distance ▾ to ping trig P0 ▾
                          echo P1 ▾
                          unit cm ▾
    if distance ▾ > ▾ 0 and ▾ distance ▾ < ▾ 20 then
        digital write pin P11 ▾ to 1
        show number 3
        set servo P2 ▾ angle to 90 °
        pause (ms) 5000 ▾
    ⊕
    if distance ▾ > ▾ 19 and ▾ distance ▾ < ▾ 40 then
        digital write pin P12 ▾ to 1
        show number 2
        set servo P2 ▾ angle to 45 °
        pause (ms) 2000 ▾
    ⊕
    if distance ▾ > ▾ 39 or ▾ distance ▾ < ▾ 1 then
        digital write pin P13 ▾ to 1
        show number 1
        set servo P2 ▾ angle to 0 °
    ⊕
    pause (ms) 1000 ▾
```

Notes/Examples

Note:

It is important that learners plan their projects.

Planning should include identifying resources required as well as planning the code.

If a project is not planned well before building it and writing code to realise the artefact, it can lead to difficulties when they need combining components and structures, as they may not have a clear roadmap for how different components and structures should work together.

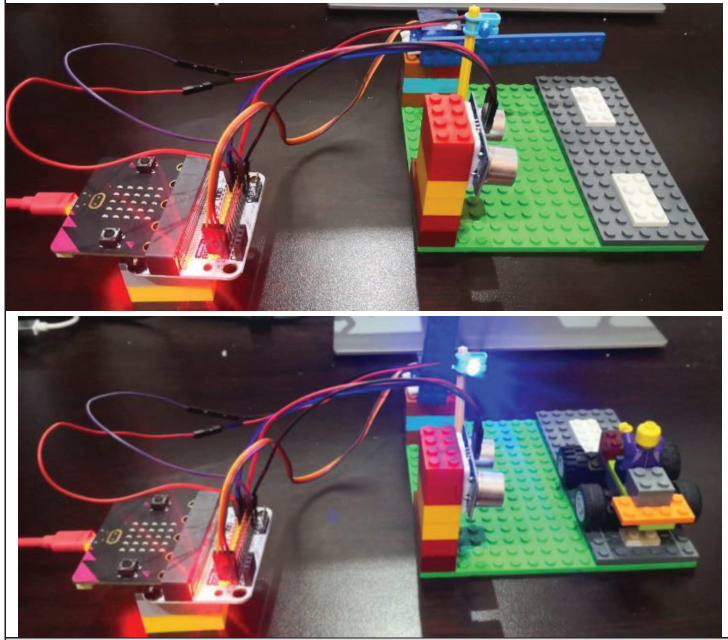
Content (Grade 8 / Term 1)

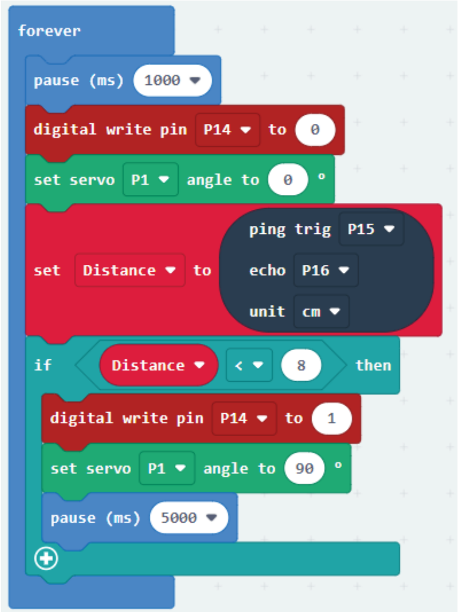
Example 2 – Boom gate

In the following example an expansion board is used which allows one to connect more components to the microcontroller.

In the example an Ultrasonic sensor, LED and Servo is used in a boom gate application. The gate opens if the ultrasonic sensor picks up a “car” in proximity and the LED is switched on. The LED switches OFF and the gate closes if there is nothing close to the proximity of the sensor.

NOTE: The learners should be guided on which pins to use.





```
forever
  pause (ms) 1000 ▾
  digital write pin P14 ▾ to 0
  set servo P1 ▾ angle to 0 °
  set Distance ▾ to ping trig P15 ▾
                         echo P16 ▾
                         unit cm ▾
  if Distance ▾ < ▾ 8 then
    digital write pin P14 ▾ to 1
    set servo P1 ▾ angle to 90 °
    pause (ms) 5000 ▾
  ⊕
```

Notes/Examples

R.7 Create, test and execute a set of robotic instructions

Example activity Microcontroller Dice Simulation Activity

Objective:

- Simulate the roll of four dice using a microcontroller. After displaying each die value, calculate and display the total value.

  Microcontroller, Display (LED matrix, LCD screen, or serial monitor)

- Set up your microcontroller with the necessary components (dice and display).
- Generate a random number between 1 and 6 for each die (simulating a roll).
- Display the value of each die on the chosen display (e.g., LED matrix or serial monitor).
- Calculate Total:
- Display the total value.

Link to R.6 and C.1 – C.7

Note:

Learners need to transfer the programming knowledge and skills acquired in the block-based coding environment (use their experience with another environment) to the new micro controller block-based coding environment.

It is a good idea to quickly revise the knowledge and skills required for the new coding environment by linking it to the first coding environment learned and explain how it works in the new environment.

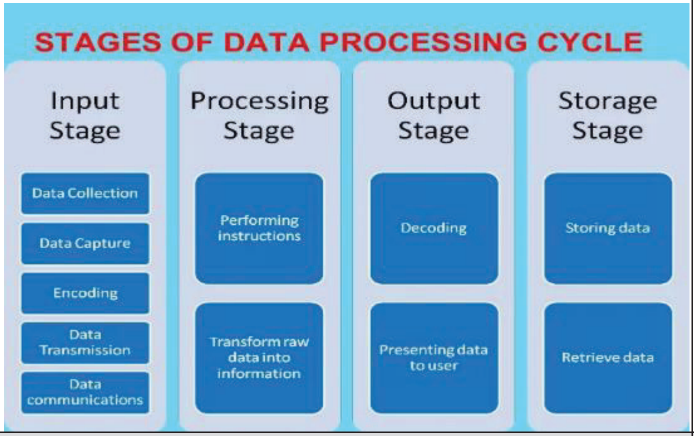
Digital Concepts	
D.1 Outline the concept of technology and purpose of information technology (IT)	**Link to D.2, D.3, and D.7**
Example activity – Why do we need computing devices In small groups learners watch the video https://youtu.be/qfUZBKDh9BY and use a KWLS chart. Learners discuss what computing devices are, the need for computing devices and how it helps them Computing devices are super helpful because they can do so many jobs fast and accurately. They help us with schoolwork, entertainment, communication (like sending messages or making video calls), and even controlling other devices like smart lights or thermostats in our homes.	Revise and extend from previous grades Learners should be able to: • •Explain what a computing device in the context of information technology is • •Relate the concept of a computing device to that of an IT/ICT tool • The role of hardware and software (apps and operating system) • The role of IT professionals in transforming data into information for making informed decisions
D.2 Recognise that he or she is living as citizens in a digital world.	**Link to D.4 and D.6 and D.10**
Finding information online. Evaluating information (false/fake news, evaluate information). **Check the source** – is it a reputable website/source> **Verify the author** – who crated/wrote the information? Is the person/organisation an authority on the topic? **Check for citations** – reliable sites usually cite their sources and provide references/links to other sites to check information **Check the publication date** – ensure that information is up-to-date and relevant, especially in technology fields that evolve rapidly **Example activity** Divide learners in small groups. Provide each group with printouts from two websites (one reputable and one scaly). Groups to evaluate both websites and prepare a short presentation based on their evaluation. Randomly selected groups present their evaluation and teacher leads discussion on the presentations.	Learners need to • Develop strong information evaluation skills to become critical consumers of online information and news • Fact-check online information to identify false/fake news and to fact-check before accepting that it is true • Demonstrate knowledge of types of false information/fake news **False news or Fake news** is news or stories created to deliberately misinform or mislead readers and includes. ○ ***Disinformation*** - Deliberately false information ○ ***Misinformation*** - Erroneous or incorrect information. Not always be deliberate; it is just wrong or mistaken. ○ ***Propaganda*** - Information, ideas, or rumors deliberately spread widely to help or harm a person, group, movement, institution, nation, etc.
D.3 Demonstrate an understanding of the concept of a computing device.	**Link to D.1, R.3, R.**

Components of a simple home network
• Group of two or more connected computers or devices (e.g. printer) that
  • are confined to small geographical area, e.g. home or small office
  • Share common communication medium such as wired or wireless connections
  • are connected to a central connecting device such as a router
  • uses a modem to connect to the internet
 	
Identify and give examples of different connectivity methods in a real-world scenario (Bluetooth, Wi-Fi, cellular, cabled).

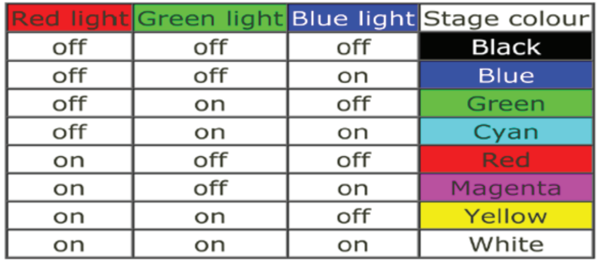
	Learners need to
	Identify the components (computing devices, network devices and communication (wired and wireless) of a basic home network
	
	A Network is
	• two or more computers and other hardware devices linked together through communication channels (wired/wireless).
	• The connection allows devices to share data.
	• It is used to share electronic communications and resources, such as printers with various users and files printers with various users

D.4 Identify the common uses of ICT in the real world.	
Examples to be discussed in small groups Divide learners into small groups. Each group draw a topic and a sub-topic, discuss uses and impact and prepare a presentation to present their discussions and findings to the class. **Entertainment and media** Streaming Services: ICT supports the delivery of music, movies, TV shows, and video games over the internet Digital Publishing: Newspapers, magazines, and books are available in digital formats, accessible on various devices Social Networking: Platforms like TikTok and Snapchat provide avenues for user-generated content and social interaction **Education** E-Learning: ICT enables online courses, virtual classrooms, and distance education, making learning accessible to people worldwide Digital Resources: Students and teachers use digital textbooks, educational websites, and multimedia tools for research and learning **Agriculture** Precision Farming: ICT tools like GPS, drones, and sensors help farmers monitor crops, optimize irrigation, and manage resources efficiently Weather Forecasting: Advanced weather prediction systems assist farmers in planning and risk management	Learners need to demonstrate knowledge of ICTs in everyday life.
D.5 Differentiate between the components of an ICT system.	
Discuss the specifications of hardware for a given computing device (Reference from advertisements). Learners differentiate between the different types of hardware components, such as input/output devices, storage devices, processors and network/connection devices, and explain their functions in a computing system. Expand on the concept of networks and introducing controlling devices remotely (e.g. cell phone as controlling device (Bluetooth speakers) or hotspot for connecting to the Internet	Revise and extend
D.6 Explain how the adaptation of technology impacted the world we work and live in	**Link to D.2 and D.3**

• Computers and networks can be used to commit crime → computer/cybercrime
• **Cyber/computer crime** is any criminal activity that involves a computer, networked device or a network
  • Done against computers or computing devices directly to damage or disable them
  • Spread misinformation or malware using computers or networks
  • Some cybercrimes do both → target computers to infect them with malware which is then spread to other machines and, sometimes, entire networks.
https://youtu.be/dRKMIG0KTkY
	Cybercrime includes
	• Steal or alter data
	• Steal personal data/information such as passwords
	• To gain unlawful use of computers or services or access to networks
	• Create and distribute malware

D.7 Present a basic understanding of the concept of input processing and output.	**Link to C.1 – C.5**
Discuss the information processing cycle regarding a given real-life scenario (input, processing, output, storage, communication). **Data as input** **Processed data** results in information that can be simulated using graphs and impact decisions   	
D.8 Interpret a pattern to represent or communicate a message or image	**D.8 and D.9 done together and Link to C.1**
D.9 Create a pattern to represent or communicate a message or image.	

Example activity – Use binary representation to create a table to represent the following:



Red light	Green light	Blue light	Stage colour
off	off	off	Black
off	off	on	Blue
off	on	off	Green
off	on	on	Cyan
on	off	off	Red
on	off	on	Magenta
on	on	off	Yellow
on	on	on	White

Extend for different patterns, e.g. words, sentences etc.

Example:

Red (R)	Green (G)	Blue (B)	Colour
0	0	0	Black
0	0	1	Blue
0	1	0	Green
0	1	1	Cyan
1	0	1	Magenta
1	1	0	Yellow
1	1	1	White

Note

Learners will need guidance from the teacher on this activity

Note:

Ensure that learners understand that binary only have two states:

off (0) and **on** (1) or true and false

Let them think of a digital system as a series of tiny switches that can be either on or off, like a light switch.

In the digital world, everything is represented using these on-off switches. Each switch is like a tiny piece of information, called a "bit." When you put many bits together, you can represent more complex things, like numbers, letters, pictures, and sounds.

D.10 Demonstrate a basic proficiency in the application of digital skills.	**Link to C.1 – C.7 and R.5 – R.7**
Introduce learners to a presentation application such as PowerPoint. (PP) – elementary level only Learners explore the PP environment and create a simple PP about themselves.	Revise file and Folder management: This PC Introduce presentation software (elementary level)

3.2.2 Term 2

Content (Grade 8 / Term 2)	**Notes/Examples**
Coding	
C.1 Apply computational thinking (CT) skills to develop a set of logical instructions to solve a problem.	**Link to C.1 – C.7**
Example activity Divide learners into pairs. Each pair solves the following problem using computational thinking. **Problem**: You need to find a picture on the Internet which you want to use as a sprite in your block-based application. Use computational thinking to describe the steps you will follow (develop an algorithm) to realise this. Take note of the characteristics of a good algorithm when designing your instructions. Pairs swop instructions/algorithms and test to see if the other pair's instructions work correctly. They now act a s the computer and follows the instructions to the letter. Each pair adds comments for the other pair, if necessary, to fix or improve their instructions. Pairs return turn the instructions with comments to the original pair to fix or improve if necessary. **Remember**: An **algorithm** is an ordered list of well-defined steps or instructions that you can follow to perform a task or solve a problem. These instructions are often expressed as something that humans can understand. Eventually, these instructions must be translated into a sequence of computer instructions using enough detail so the computer can execute the instructions. **Developing an algorithm** To develop an algorithm, you must identify what needs to be done (the instructions) and the order in which they must be done. When developing an algorithm, each instruction is identified and the sequence in which the instructions are carried out, is planned and must be logical.	**Remember** **Sequence and detail** **Sequencing** is putting events or information in a specific order. It is the skill that helps you to plan ***what*** steps to take ***to*** perform a task successfully. **Detail** means considering every aspect or minor part of something. It is to describe or give exact information about something. The steps or instructions to perform a task need to be unambiguous – they need to be precise and clear to avoid misinterpretation or different interpretations by different people. **How are sequence and detail helpful?** In **sequencing**, we learn about patterns in relationships. We also learn to understand the order of things. By learning to sequence, we develop the ability to understand and arrange purposeful patterns of actions, behaviours, ideas, or thoughts. Attention to **detail** is important because it helps prevent mistakes and ensures the successful completion of a task.
C.2 Present a simple coding solution using symbolic or written statements representing sequences of commands, single repetition, and conditional constructs.	**Link to C.1 C.3 – C.7**

Content (Grade 8 / Term 2)

Example activity 1

You want to explain to your younger brother how to determine whether you pass a test. You wrote down the following main steps:

1. Get the test mark (mark obtained for test)
2. Get the test total (Total marks for test)
3. Calculate the percentage
4. If percentage is more than 30, you have passed, else you have failed

Study the following IPO table and determine if the algorithm is correct

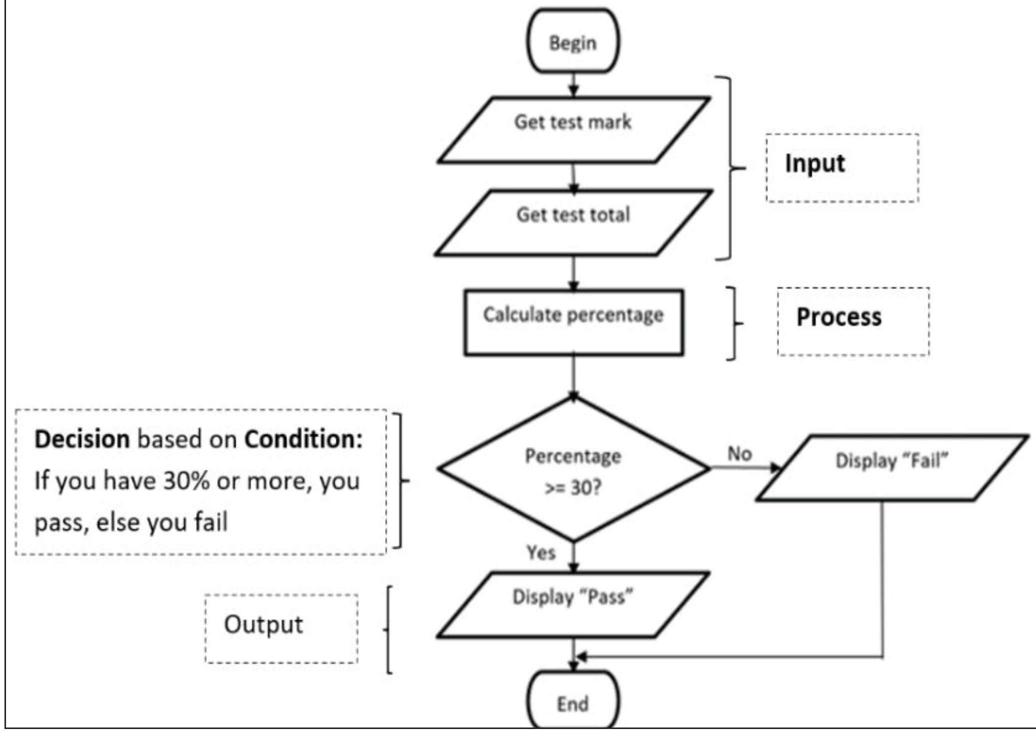
You can represent the above instructions/algorithm using a flowchart:

Input	Process	Output
Test mark	Calculate percentage	Display pass if percentage >= 30 or
Total of test	Determine if percentage is greater or equal to 30	Display if percentage is < 30

Example activity 2

Convert the flowchart into block-based code.

Run the program, test and debug

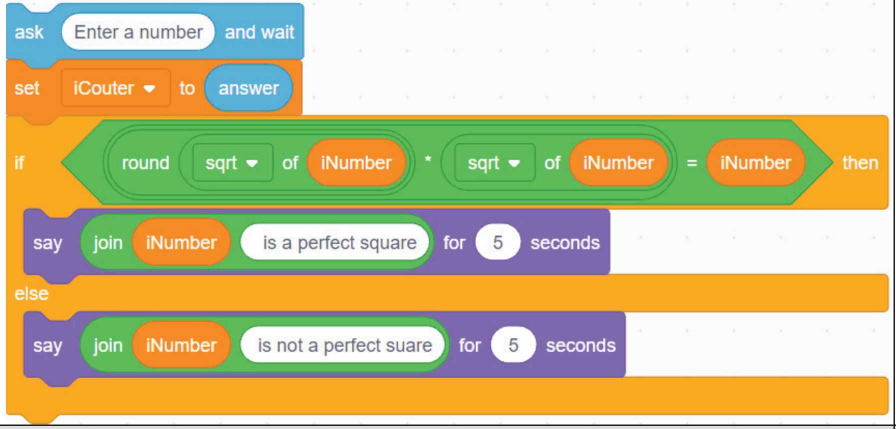


Notes/Examples

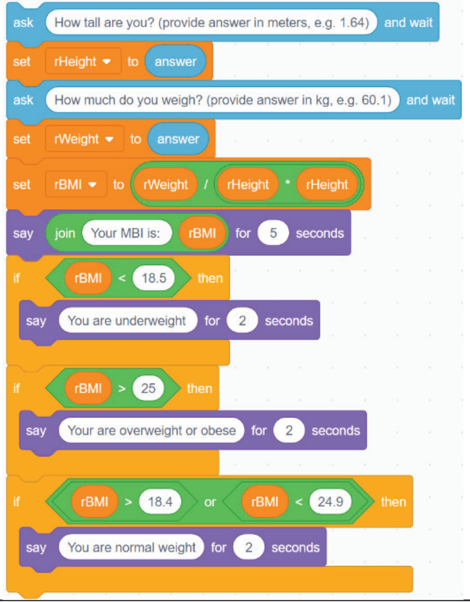
Note:

Explain the symbols in the table below to the learners.

Symbol	Action	Description/Function
⬭	Begin / End (Start / Stop)	An oval represents the beginning or the end of instructions
▱	Input / Output (Read / Write)	A parallelogram represents the input (data) or output (results) of the solution
▭	Processing	A rectangle represents a process, e.g. calculation
◇	Decision	A diamond represents a decision based on a condition (Allows for taking different paths)
↓	Flow of instructions (Next instruction)	An arrow is a connector between the symbols that shows the relationship between the symbols as well as the flow and order of the instructions

Content (Grade 8 / Term 2)	Notes/Examples
C.3 Interpret and execute a given symbolic or written set of commands	**Link toC.4**
Example activity – Trace code to explain what it does Use a trace table to work through the following program and explain what it does 	**Note:** It is important that coding activities revise coding concepts learned in previous terms and grades cumulatively, using different activities and combinations of concepts. **Note:** Literature suggests that the biggest problem of novice programmers does not seem to be the understanding of basic coding concepts but rather learning to **apply** them. Therefore, at this level, beware of giving learners programming tasks that combine too many concepts (Robins, 2019).

```
ask (Enter a number) and wait
set iCouter ▾ to (answer)
if <(round ((sqrt ▾ of (iNumber)) * (sqrt ▾ of (iNumber)))) = (iNumber)> then
  say (join (iNumber) (is a perfect square)) for (5) seconds
else
  say (join (iNumber) (is not a perfect suare)) for (5) seconds
```

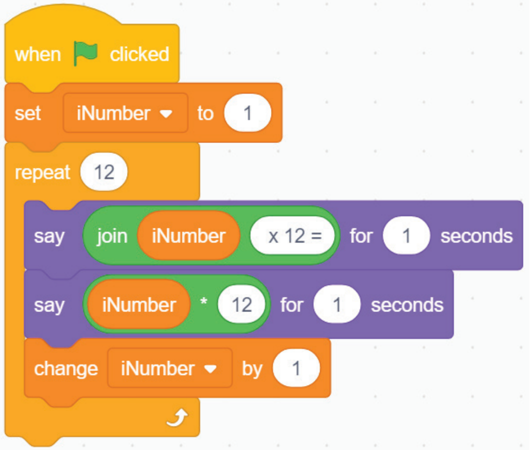
Content (Grade 8 / Term 2)	Notes/Examples
C.4 Debug a given symbolic or written set of instructions	**Link to C.3**
Example activity – BMI calculator To determine if one is overweight, on can calculate one's Body Mass Index (BMI) BMI is calculated through the following formula: $BMI = \frac{Weigth\ (kg)}{[Height\ (m)]^2}$ The BMI is interpreted as follows: • Underweight = <18.5 • Normal weight = 18.5–24.9 • Overweight = 25–29.9 • Obesity = BMI of 30 or greater Get a person's weight (in kg) and height (in m), then determine if the person is overweight or not. Display a message to that will inform a person if he/she is underweight, normal weight, overweight or obese. Someone wrote a program to determine a person's BMI, but it does not work correctly. Use a trace table to work through the program and determine what the problem is. Correct the code and execute the code to test it. If it works correctly, WELL DONE, else repeat the process to find the problem until the program works correctly. 	**Note:** Learners must understand the difference in the use of AND OR (link to truth tables done with Rs) Debugging is an integrated and important part of coding. While learners need to debug all the code the write, learners must also be provided with incorrect code which they need to debug.

```
ask (How tall are you? (provide answer in meters, e.g. 1.64)) and wait
set rHeight ▾ to (answer)
ask (How much do you weigh? (provide answer in kg, e.g. 60.1)) and wait
set rWeight ▾ to (answer)
set rBMI ▾ to ((rWeight) / ((rHeight) * (rHeight)))
say (join (Your MBI is: ) (rBMI)) for (5) seconds
if <(rBMI) < (18.5)> then
  say (You are underweight) for (2) seconds
if <(rBMI) > (25)> then
  say (Your are overweight or obese) for (2) seconds
if <<(rBMI) > (18.4)> or <(rBMI) < (24.9)>> then
  say (You are normal weight) for (2) seconds
```

Content (Grade 8 / Term 2)	Notes/Examples
C.5 Evaluate a given solution towards potential improvement.	
Example activity – Multiplication table The following two programs are provided. Both programs can display the 12-times table Study both algorithms and describe how each works. Evaluate each. Which algorithm is the best? Explain why.	**Note:** Many learners tend to focus on very small parts of the code and lose sight of the "big picture".

Content (Grade 8 / Term 2)

Program 1



```
when 🏴 clicked
set iNumber ▾ to 1
repeat 12
    say join iNumber x 12 = for 1 seconds
    say iNumber * 12 for 1 seconds
    change iNumber ▾ by 1
```

Program 2

```
when 🏴 clicked
set iNumber ▾ to 1
ask Which table do you want to leearn? and wait
set iTable ▾ to answer
ask How many multiples do you want? and wait
set iMultiple ▾ to answer
repeat iMultiple
    say join iNumber join x iTable for 1 seconds
    say iNumber * iTable for 1 seconds
    change iNumber ▾ by 1
```

Notes/Examples

They are also prone to focus on superficial aspects of the task/problem that are not functionally central to the solution (Lister & Teague, 2014)

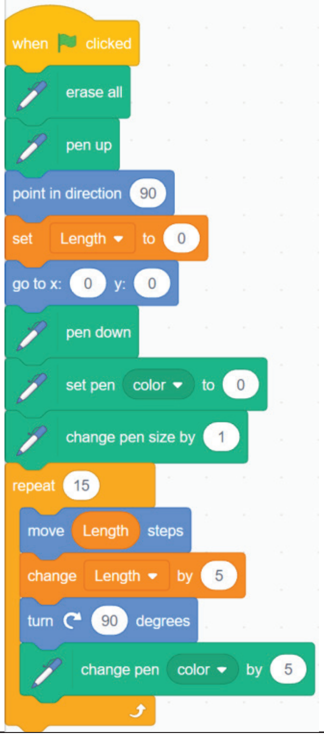
C.6 Recognise and interpret patterns in symbolic sets of data or visualisations.

Example activity – Interpret a pattern

Divide learners into pairs.
The driver receives the code on the right on paper
The navigator receives a blank page and a pen/pencil.
The driver reads the code, calls out the code and acts out the instructions (the driver may ignore acting out the pen instructions but still calls out the pen instructions for the navigator who holds the pen/pencil)
As the driver calls out and acts out an instruction, the navigator draws the instruction on paper.
The navigator must also hold count of the number of times a loop is executed and must tell the driver when to exit the loop.
Both the driver and the navigator must also hold count of the variable (length) contents to ensure the instructions are carried out correctly.
When done, the driver copies the code to a block-based coding app, runs the code on the computer and the navigator compares the result with the drawing made during acting out process.

If the navigator's drawing is incorrect, the pair need to determine where they misinterpreted the code, or they can switch roles and repeat the exercise to determine where they went wrong.

The pair then change the code as follows:
Replace the no 4 with no 3 in both loops
Replace the 90 degrees with 120 degrees in both instances.
Run the code and watch what happens
Refer to the activity in Term 1 C.5 and play with different number of repeats and degrees.



```
when 🏴 clicked
erase all
pen up
point in direction 90
set Length ▾ to 0
go to x: 0 y: 0
pen down
set pen color ▾ to 0
change pen size by 1
repeat 15
    move Length steps
    change Length ▾ by 5
    turn ↻ 90 degrees
    change pen color ▾ by 5
```

Link to C.1, C.2 and D.6 and D.7L

You want to interleave practise in problems.

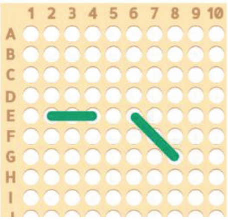
"It is important that problem types must differ, for example, you want to randomly have a problem of one type and then solve a problem of another type and then a problem of another type. And in doing that, it feels difficult, and it doesn't feel fluent. And the signals to your brain are, I am not getting this. I am not doing very well. But in fact, that effort to try to figure out what kinds of approaches do I need for each problem as I encounter a different kind of problem, that is producing learning. That is producing robust skills that stick with you."

Dr Mark A. McDaniel, Harvard University

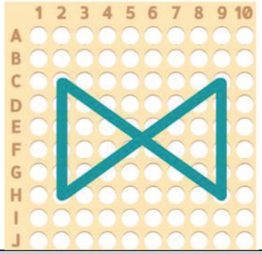
Content (Grade 8 / Term 2)	Notes/Examples
C.7 Create or complete a pattern to represent a data set	

Example activity - Create a pattern

An embroidery machine uses the following commands for creating patterns:
OUT (cc)-IN (dd), where cc and dd indicate the position of the needle in the grid.
Example:
OUT(E6)-IN(G8) moves the needle to the E6 position and pulls the cotton thread through the fabric from behind. The needle then moves to the G8 position and pushes the thread from the front through to the back. The following two commands create a pattern like the one below: OUT(E6)-IN(G8); OUT(E2)-IN (E4
Create an embroidery pattern by programming instructions for his embroidery machine.



Write down the set of commands that will complete the following pattern
2020-Talent-Search-Solutions-book.pdf (olympiad.org.za)



Note:
This task is an example of how an algorithm can be used to create a pattern by an embroidery machine.
An algorithm describes the steps we follow to complete a task. Algorithms are common in computer science, but outside computer science algorithms may play a role in solving **everyday** life problems.

By identifying patterns, we can predict what will come next and what will happen again and again in the same way.
In Computer Science/coding we analyse patterns in data and make predictions and generalisations based on the pattern analysis.

Robotics

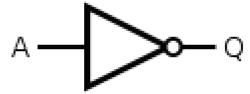
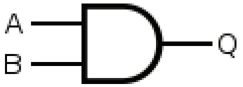
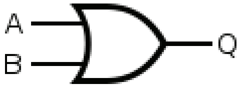
R.3 Outline the different components of a robot

Introduce the RGB LED in detail.

Component	Basic definition	Purpose	Practical application example – May be used to:
RGB LED (Red, Green, Blue LED): 	An LED capable of emitting different colours by varying the intensity of its red, green, and blue components.	Used for colour indication, lighting effects, and display purposes in robots.	Used to display colours or convey information through different colour combinations.

Boolean Logic:	Logical Gates:
Boolean logic is a fundamental branch of mathematics and logic developed by George Boole in the mid-19th century. It deals with true and false values, represented as 1 and 0, respectively. In Boolean logic, statements are evaluated based on these binary values, and operations are performed using logical operators such as AND, OR, NOT, etc. These logical operations are fundamental building blocks in decision-making and control processes	Logical gates are electronic circuits that implement Boolean logic operations. They take one or more binary inputs and produce an output based on the predefined logical function. The three most basic logical gates are AND, OR, and NOT gates. Complex logical operations can be constructed by combining these basic gates.

The learners are introduced to the 3 ANSI symbols for the three logical gates.

NOT	AND	OR
		

Also see (https://en.m.wikipedia.org/wiki/Logic_gate
https://www.computerscience.gcse.guru/theory/logic-gates

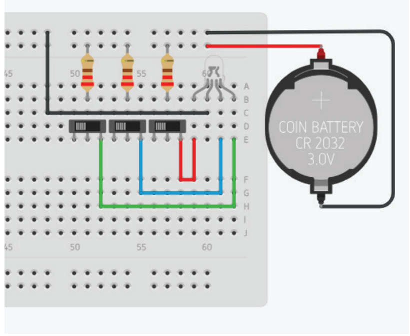
Introduce Boolean logic and logical gates.

Learners understand the principles of
- Boolean logic
- Logical gates (AND, OR and NOT)
- How it relates to the design, programming and functioning of robotic systems

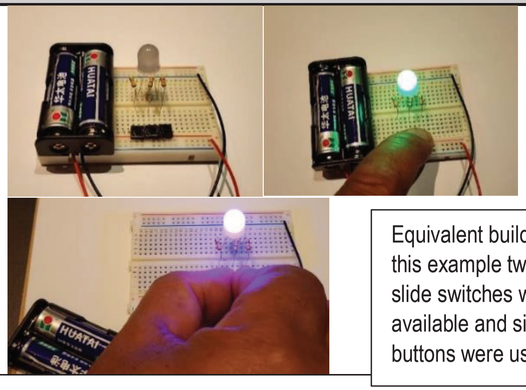
Learners can
- Represent a simple truth table for the AND, OR and NOT gates
- Interpret a simple logical gate diagram provided and comprising of no more than 3 inputs and a maximum of four gates to determine a single outcome

R.5 Design a simple artefact based on a set of design instructions	**Done with R.6 and R.7**
R.6 Mimic the operations of a robot	**R.5 and R.6 done together**
Breadboard project – RGB LED **Outline how an RGB LED works and have the learners understand how the various colours are displayed. Links to truth tables.**	Reinforce concepts pertaining to components.

Content (Grade 8 / Term 2)	Notes/Examples

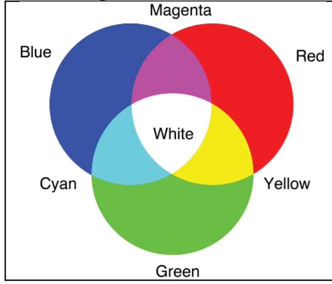


Note: Breadboard design (given to learners)



Equivalent build by learner (In this example two position slide switches was not available and simple push buttons were used)

The mixing of colours and the truth table



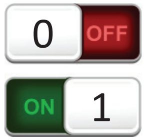
R	G	B	Result
0	0	0	0
0	0	1	🟦
0	1	0	🟩
0	1	1	🟦
1	0	0	🟥
1	0	1	🟪
1	1	0	🟨
1	1	1	⬜

Project 2 (Micro: bit + RGB)

Digital and analogue pin output

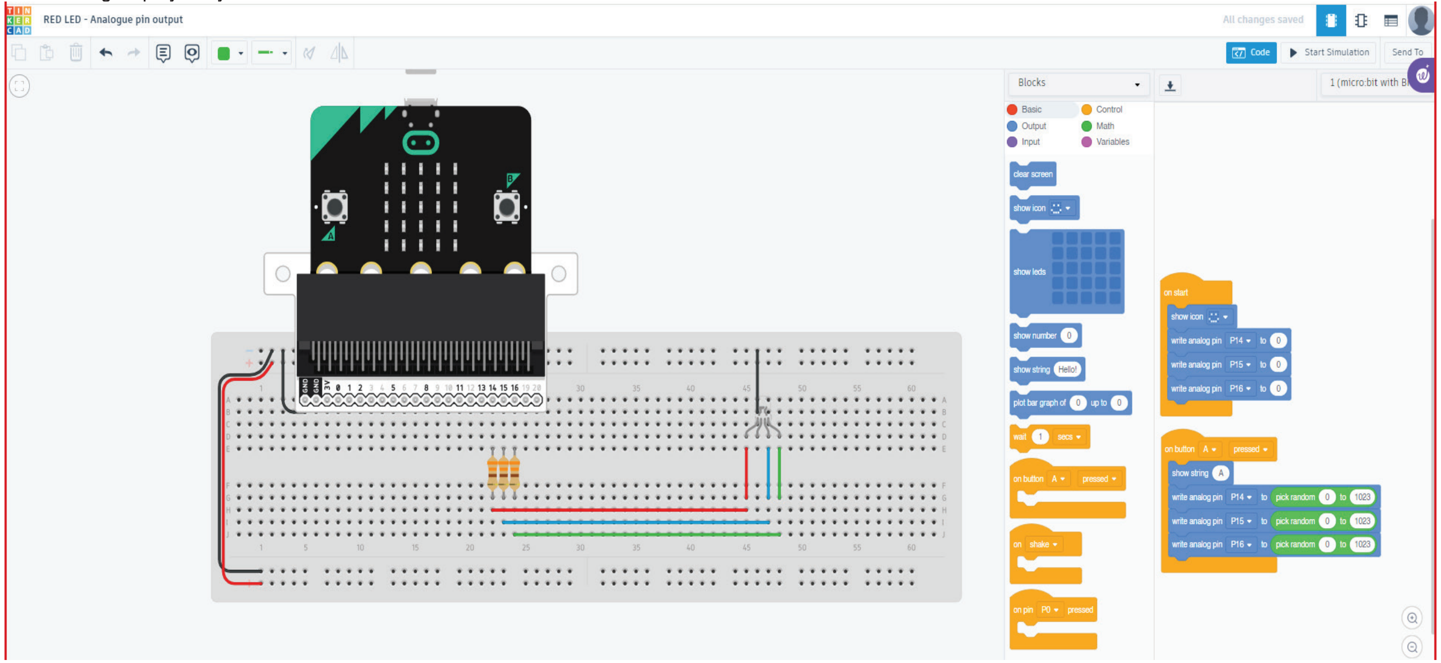
Introduce the learners to the concept of digital pin vs analogue pin output.

Differentiate between digital and analogue pin output and how it is used in robotics or physical computing applications.

	Basic Definition	Purpose	Usage in Robotics, in respect of a (Microcontroller)
Digital pin output 	Digital pins on a microcontroller are used for sending digital signals, which can have only two possible states: HIGH (1) or LOW (0).	Digital pins are primarily used for tasks that involve binary on/off control, like turning an LED on or off, controlling a motor's direction, or sending digital signals to other digital devices.	In robotics projects with a microcontroller, digital pins can be used to control various components such as LEDs, servos, motors, and relays. For instance, you can use a digital pin to turn on or off an LED to indicate the robot's status or activate a motor for a specific robotic movement.
Analogue pin output 	Analog pins on a microcontroller can output analogue signals, which provide a range of values between 0 and 1023, representing different levels of voltage or intensity.	Analog pins are typically used for tasks that require variable or smooth control, such as dimming the brightness of an LED, controlling the speed of a motor, or reading analogue sensor values.	In robotics, analogue pins on a microcontroller can be used to control the speed of a motor, adjust the position of a servo, or read values from analogue sensors like light-dependent resistors (LDRs) or potentiometers. For example, you can use an analogue pin to read the output from a distance sensor and adjust the robot's speed or behaviour based on the detected distance.

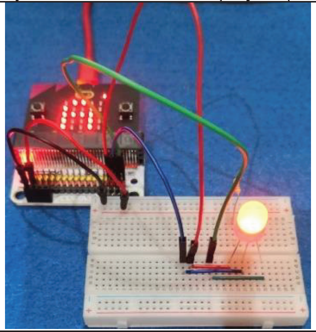
Content (Grade 8 / Term 2)

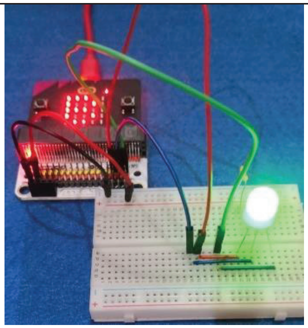
In the following simulated example random values are chosen to be passed as analogue values to each of the pins to an RGB LED resulting in a different random colour being displayed by the LED.

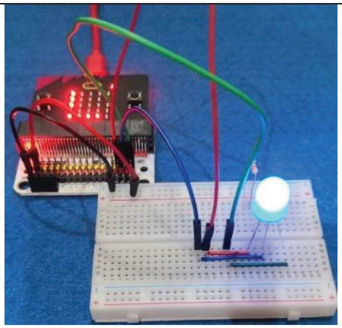


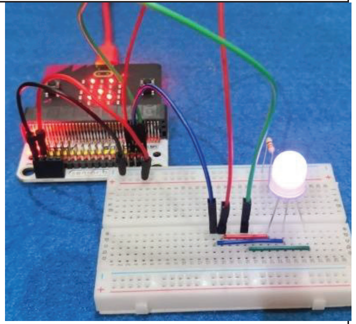
```
on start
    show icon
    write analog pin P14 to 0
    write analog pin P15 to 0
    write analog pin P16 to 0

on button A pressed
    show string A
    write analog pin P14 to pick random 0 to 1023
    write analog pin P15 to pick random 0 to 1023
    write analog pin P16 to pick random 0 to 1023
```

Physical build of the RGB project (The R

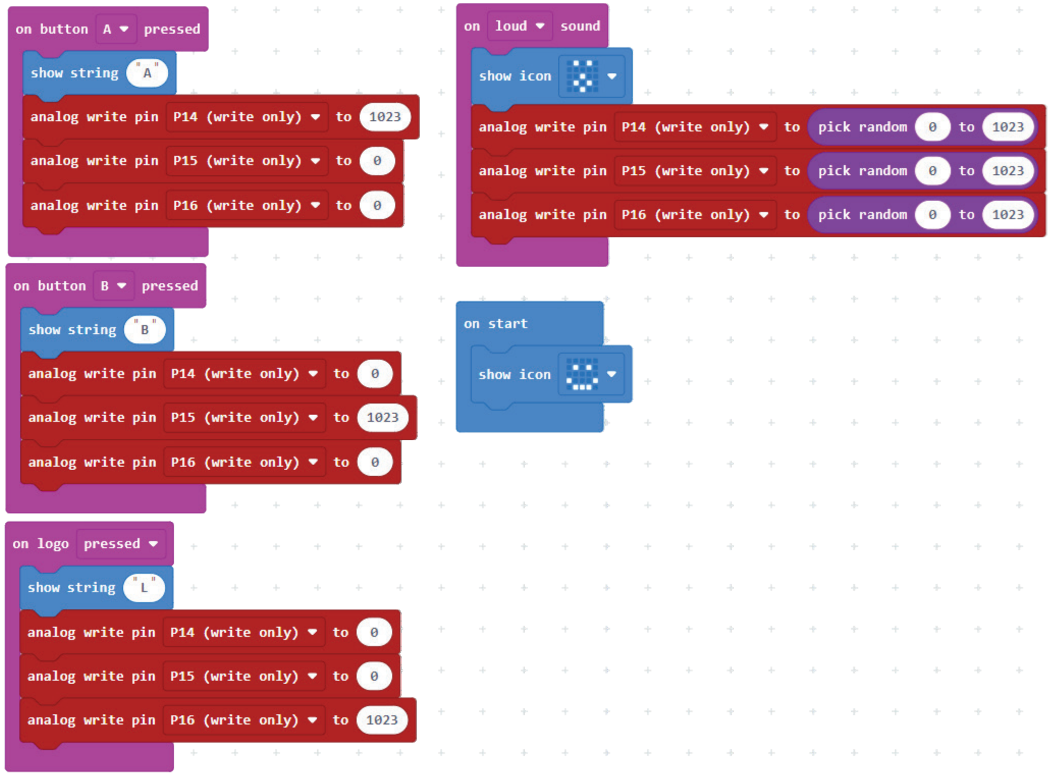






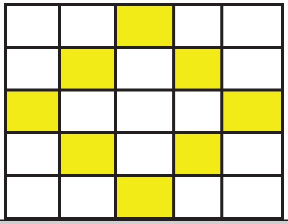


Notes/Examples

Content (Grade 8 / Term 2)	Notes/Examples
	

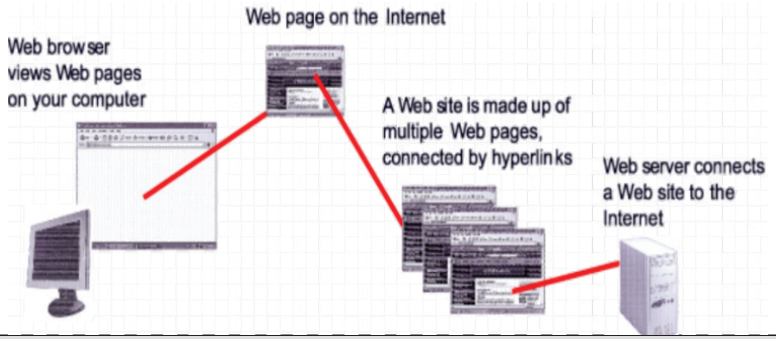
```
on button A ▾ pressed
  show string "A"
  analog write pin P14 (write only) ▾ to 1023
  analog write pin P15 (write only) ▾ to 0
  analog write pin P16 (write only) ▾ to 0

on button B ▾ pressed
  show string "B"
  analog write pin P14 (write only) ▾ to 0
  analog write pin P15 (write only) ▾ to 1023
  analog write pin P16 (write only) ▾ to 0

on logo pressed ▾
  show string "L"
  analog write pin P14 (write only) ▾ to 0
  analog write pin P15 (write only) ▾ to 0
  analog write pin P16 (write only) ▾ to 1023

on loud ▾ sound
  show icon ▾
  analog write pin P14 (write only) ▾ to pick random 0 to 1023
  analog write pin P15 (write only) ▾ to pick random 0 to 1023
  analog write pin P16 (write only) ▾ to pick random 0 to 1023

on start
  show icon ▾
```

R.4 – Present an understanding of how robots affect the world.	**Link to R.1, R.2 and R.3**
Example activity –How they affect the world Provide learners with videos to watch, e.g. https://youtu.be/_Lad2u931Cw and https://youtu.be/l4iW-L5J7dg and a KWLS chart. Learners watch videos and complete the KWLS chart Now, divide learner into pairs. Each pair identify a field/type of robots, e.g. the hospitality industry, discuss advantages/disadvantages and what ethical dilemmas might occur	The learners should be able to: • present a basic understanding and consider the potential ethical dilemmas related to robotic technologies, such as privacy concerns, AI bias, jobs, and the responsibility for actions performed by autonomous robots. • present a basic description and understanding of robot programming and its role in automation and how it plays a significant role in robot functionality and their integration into various industries.
R.7 Create, test and execute a set of robotic instructions	**Link to R.5 and R.6**
Example activity 1 – Wave illusion Display any picture or pattern and create the illusion of a wave moving through the pattern. When Button A is pressed, a "wave" must move through the pattern from left to right. Tip: Change the intensity of the first column of LED lights for 500 ms, then for the second column, until it is done for all columns. When Button B is pressed, a "wave" must move through the pattern from right to left.	**Note:** Learners need to transfer the programming knowledge and skills acquired in the block-based coding environment (use their experience with another environment) to the new micro controller block-based coding environment.

Content (Grade 8 / Term 2)	Notes/Examples
Example activity 2 “catch the dot” game. The purpose of this game is to navigate a basket (LED bar) to “catch” other LED lights. Press both buttons to start the game. Create a “basket” or “bar” that is 3 LED lights long. The basket displays at the bottom of the screen when the game starts. Display a random dot on the microcontroller for a random number of seconds. Move the basket with up-, down-, left- and right arrows to “catch” the dot. Repeat the game 10 times (or as many times as you want to). When Button A is pressed, display the number of dots that you caught with the basket. **Example activity 3 “reveal the picture”** The purpose of this program is to create a continuously flashing picture by revealing a pattern or picture starting from left to right. The learner must use loops to switch on the lights and not design different grids manually. Reveal the LED lights starting from the first column until the whole pattern is displayed. Use a loop to display (switch on) the LED lights from the first column to the last column. Repeat the process to make a “flashing” pattern. See an example of a pattern on the right. You can also design your own pattern. 	It is a good idea to quickly revise the knowledge and skills required for the new coding environment by linking it to the first coding environment learned and explain how it works in the new environment.
Digital Concepts	
D.2 Recognise that he or she is living as citizens in a digital world.	
Example activity – Discuss the POPI Act and copyright issues Discuss what the POPI Act does: **Keep information safe** - The POPI Act makes sure that companies and organizations, like schools, banks, and websites, must keep your personal information safe. They can't just share it with anyone **Ask for permission** - Before a company can use your personal information, they must ask for your permission. It's like asking for your permission to read your secret diary **Tell you what they are doing** - Companies must explain to you why they need your information and what they'll do with it. It's like them telling you why they want to read your diary **Let you control your information** - The POPI Act gives you the power to say "no" if you don't want a company to use your information. It's like you can decide who can or cannot read your diary **Say what happens if they do not follow rules** - If a company doesn't follow the rules in the POPI Act and doesn't keep your information safe, they can get into trouble and must pay a fine. **Examples**: • Say you sign up for a game app on your phone. The app company must ask your mom or dad for permission because you're a Grade 7 learner, and they need to explain why they want your information (like your name and age) and how they'll use it (to create your game account). If they don't ask for permission or share your information without permission, they can get in trouble • At school, your teacher keeps a list of your names and marks. They can't just show that list to everyone. They must keep it safe and only share it with people who need to see it, like your parents The **POPI Act** is like a protector of your personal information. It makes sure that your information is treated carefully and respectfully by companies and organizations. Just like you want your secret diary to be safe and private, the POPI Act helps keep your personal information safe and private tool Now, divide learners into small groups. Provide each group with information on POPIA as well as copyright with questions they need to answer. Groups report on the following: • What the POPI act is • What copyright is • Why sources should be referenced • How to protect personal information	Learners need to understand what copyright and plagiarism is (about software, information, intellectual property), including • Awareness of POPIA (POPI Act) • What copyright is • How to protect personal information • What the Public Domain is • Awareness that sources should be referenced • Understand the significance of network security and the importance of protecting personal and sensitive information.

<table>
<tr><th>Content (Grade 8 / Term 2)</th><th>Notes/Examples</th></tr>
<tr><td>D.4 Identify the common uses of ICT in the real world</td><td>Link to D.5</td></tr>
<tr><td>

- **Website** – collection of web pages with related information
- **Webpage** is a document that have text, images, videos, animations, etc. (a page in a website)
- **URL** (Uniform Resource Locator) – A web address that is unique to each resource on the web. It could be the address of a webpage or an image file
- **Hyperlink** – links to other webpages
- **Web server** – A computer where files are stored which can be accessed via the internet using HTTP
- **Browser** – used to access websites and web pages. It allows you to see HTML document as a beautiful document e.g. Edge, Chrome. It interprets HTML code to display images, etc.
- **Search engine** – used to search for specific information on the web within the browser, e.g. Google, Bing



Example activity – Explore Web concepts

Provide learners with a KWLS table on a worksheet

Learners watch the following videos and complete the columns in the KWLS chart

https://youtu.be/X6DIH7b82Qw
https://youtu.be/AGGWCbMvOxA
https://youtu.be/-U-j88slQUk or https://youtu.be/d15SBe3diu4

</td><td>

Discuss WWW concepts

Learners need to demonstrate knowledge (what it is) of WWW concepts:

- Website
- Webpage
- Web server
- Browser
- Search engine
- URL
- Hyperlink

Learners need to provide an example of

- URL
- Browser
- Search engine
- Website

Learners need to distinguish between a website and a webpage

</td></tr>
<tr><td>D.6 Explain how the adaptation of technology impacted the world we work and live in</td><td>Link to D.2</td></tr>
<tr><td>

Example activity - Importance of the WWW

- Define the concept of the World Wide Web (WWW) and its role in the broader context of the internet.
- Explain how the WWW facilitates communication, collaboration, and information sharing among individuals and organisations.
- The world wide web opened the internet to everyone, not just scientists.
- It connected the world in a way that was not possible before and made it much easier for people to get information, share and communicate.
- It allowed people to share their work and thoughts through social networking sites, blogs and video sharing.

</td><td>

Learners need to understand

- what the WWW is
- how it facilitates communication, information sharing and collaboration

</td></tr>
<tr><td>D.7 Present a basic understanding of the concept of input processing and output.</td><td>Link to C.2 – C.5</td></tr>
<tr><td>Understand how the Information Processing Cycle relates to networks in terms of Input, output, processing, storage and communication (link to coding & robotics)</td><td>Revise information processing cycle</td></tr>
<tr><td>D.8 Interpret a pattern to represent or communicate a message or image.</td><td>D.8 and D.9 done together</td></tr>
<tr><td>D.9 Create a pattern to represent or communicate a message or image.</td><td>Link to C.5 – C.7</td></tr>
<tr><td>

Use algorithms and transfer to block-based coding

<u>Example activity</u>:

Use the following algorithm to ask for a binary number and display the corresponding button in the colour as shown in the table below.

Use the table to determine the outcome.

1. Create costumes for each colour in the table above. Number the costumes from 1.
2. Ask for a Binary number.
3. Read and store the input binary code.
4. If the first letter of the binary code entered is equal to 1, add 4 to the variable.
5. If the second letter of the binary code entered is equal to 1, add 2 to the variable.
6. If the third letter of the binary code entered is equal to 1, add 1 to the variable.
7. Switch to the corresponding costume.

<table>
<tr><th>Red (R)</th><th>Green (G)</th><th>Blue (B)</th><th>Colour</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Black</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Blue</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Green</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Cyan</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Magenta</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Yellow</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>White</td></tr>
</table>

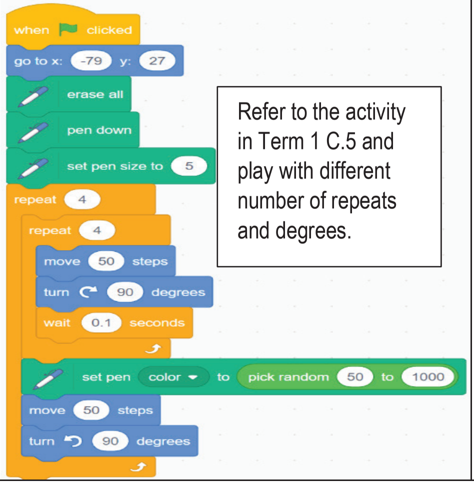
</td><td>

Possible solution:

```
when clicked
ask Binary number and wait
set Binary to answer
set my variable to 0
if letter 1 of Binary = 1 then
    set my variable to my variable + 4
if letter 2 of Binary = 1 then
    set my variable to my variable + 2
if letter 3 of Binary = 1 then
    set my variable to my variable + 1
switch costume to my variable + 1
```

</td></tr>
<tr><td>D.10 Demonstrate a basic proficiency in the application of digital skills.</td><td>Link to Cs and Ds</td></tr>
<tr><td>Revise file and folder management.</td><td></td></tr>
</table>

3.2.3 Term 3

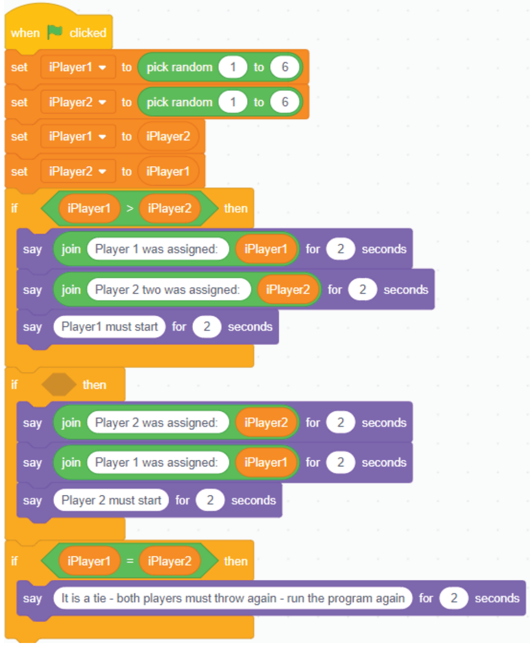
<table>
<tr><th colspan="2">Content (Grade 8 / Term 3)</th><th>Notes/Examples</th></tr>
<tr><td colspan="2">Coding</td><td></td></tr>
<tr><td colspan="2">C.1 Apply computational thinking (CT) skills to develop a set of logical instructions to solve a problem.</td><td>Link to all Cs</td></tr>
<tr><td>Example activity 1 – Solve a scrambled algorithm
Divide learners into pairs and provide each pair with a worksheet with the algorithm on the right that explains how the game snakes and ladders must be played.
Each pair studies the algorithm and read and explain the algorithm to each other. (Do pair reading. One learner starts and read the first instruction, and the pair decides on the meaning. The other learner in the pair then reads the next line and the pair decides on the meaning, etc.)
One learner in the pair cuts out the algorithm (on the lines) and the other learner in the pair scrambles the algorithm.
Pairs swop their seats with other pairs to unscramble their algorithms.
Each pair must put the algorithm back in the original order.
The pair with whom they swopped initially must check each other's solutions.</td><td><table>
<tr><td>Snakes and ladders algorithm.</td></tr>
<tr><td>Place the icon on the first block on the bottom left of the board</td></tr>
<tr><td>Repeat</td></tr>
<tr><td>Player throws a die</td></tr>
<tr><td>Repeat</td></tr>
<tr><td>Move the icon forward</td></tr>
<tr><td>Until the number of moves is the same as the number on the dice</td></tr>
<tr><td>If the icon lands on a snake's head, then</td></tr>
<tr><td>move down and place the icon on the snake's tail</td></tr>
<tr><td>else if the icon lands on the lower part of a ladder, then</td></tr>
<tr><td>move up and place the icon up to the top of the ladder</td></tr>
<tr><td>else</td></tr>
<tr><td>leave the icon where it is</td></tr>
<tr><td>EndIf</td></tr>
<tr><td>Until icon lands on the top right block</td></tr>
<tr><td>Say "You are the winner!"</td></tr>
</table></td><td><i>You want to interleave practise in problems.</i>
<i>"It is important that problem types must differ, for example, you want to randomly have a problem of one type and then solve a problem of another type and then a problem of another type. And in doing that, it feels difficult, and it does not feel fluent. And the signals to your brain are, I am not getting this. I am not doing very well. But in fact, that effort to try to figure out what kinds of approaches do I need for each problem as I encounter a different kind of problem, that is producing learning. That is producing robust skills that stick with you."</i>
Dr Mark A. McDaniel, Harvard University</td></tr>
<tr><td colspan="2">C.2 Present a simple coding solution using symbolic or written statements representing sequences of commands, single repetition, and conditional constructs.</td><td>Link to C.1, C.3 – C.7 and R.6 – R.7</td></tr>
<tr><td colspan="2">Example activity 1– Funny name generator
The purpose of the program is to swap the first letters of a name and a surname.
Example: I the name and surname are "Susan Duncan" then the name "Dusan Suncan" must be displayed.
Input: A name and a surname.
Output: A string that displays the name and surname in which the first letters were swopped around.</td><td></td></tr>
<tr><td colspan="2">C.3 Interpret and execute a given symbolic or written set of commands</td><td></td></tr>
<tr><td>Example activity – Evaluate code, explain what it does and change to achieve a different outcome
Divide learners into pairs.
The driver receives the code on the right on paper
The navigator receives a blank page and a pen/pencil.
The driver reads the code, calls out the code and acts out the instructions (the driver may ignore acting out the pen instructions but still calls out the pen instructions for the navigator who holds the pen/pencil)
As the driver calls out and acts out an instruction, the navigator draws the instruction on paper. The navigator must also hold count of the number of times a loop is executed and must tell the driver when to exit the loop.
When done, the driver copies the code to a block-based coding app, runs the code on the computer and the navigator compares the result with the drawing made during acting out process.
If the navigator's drawing is incorrect, the pair need to determine where they misinterpreted the code, or they can switch roles and repeat the exercise to determine where they went wrong.
The pair then change the code as follows:
Replace the no 4 with no 3 in both loops
Replace the 90 degrees with 120 degrees in both instances.
Run the code and watch what happens</td><td>when 🏁 clicked
go to x: -79 y: 27
erase all
pen down
set pen size to 5
repeat 4
repeat 4
move 50 steps
turn ↻ 90 degrees
wait 0.1 seconds
set pen color to pick random 50 to 1000
move 50 steps
turn ↺ 90 degrees
Refer to the activity in Term 1 C.5 and play with different number of repeats and degrees.</td><td>Note:
Generally, most learners are comfortable with tasks/problems that require them to write code that requires them to combine one or two concepts at a time. Manny learners, however, struggle when they must combine coding concepts at a time/in one program as it increases the difficulty level as well as the complexity of the task/problem. It is therefore advisable that leaners practise coding concepts using small, basic, manageable tasks/problems until they are ready for the next step.

Some learners may require guidance when the task requires several coding concepts to be combined.</td></tr>
</table>

Content (Grade 8 / Term 3)	Notes/Examples
C.4 Debug a given symbolic or written set of instructions	**Link to C,1 – C.3 andC.5 – C.7 as well as R.5 – R.7**

Example activity – Swop values

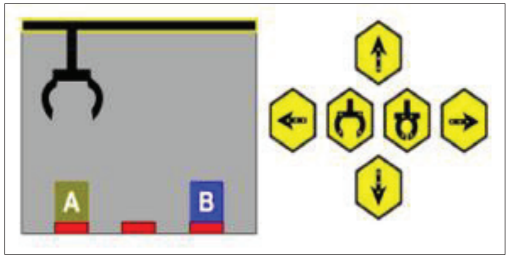
The code on the right simulates two people throwing a die to determine who should start when playing a game. The player that gets the largest number, starts the game. They use the Random instruction to simulate the throw of a dice. A random value from 1 – 6 is assigned to each player.
If the players are assigned the same number, the program tells them that it is a tie and that they must run the program again to determine who starts the game.
To make the game a little more interesting, they decide that after the random values are assigned to each player, they will first swop values (the random value assigned for Player 1 and the random value for Player 2 must be swopped), then the player with the biggest number will start the game.
The code, however, does not work correctly.
Run the code and determine what is wrong, Then, debug the code and test again.



```
when [flag] clicked
set iPlayer1 ▾ to pick random 1 to 6
set iPlayer2 ▾ to pick random 1 to 6
set iPlayer1 ▾ to iPlayer2
set iPlayer2 ▾ to iPlayer1
if iPlayer1 > iPlayer2 then
    say join Player 1 was assigned: iPlayer1 for 2 seconds
    say join Player 2 two was assigned: iPlayer2 for 2 seconds
    say Player1 must start for 2 seconds
if then
    say join Player 2 was assigned: iPlayer2 for 2 seconds
    say join Player 1 was assigned: iPlayer1 for 2 seconds
    say Player 2 must start for 2 seconds
if iPlayer1 = iPlayer2 then
    say It is a tie - both players must throw again - run the program again for 2 seconds
```

Note:
If you struggle work through the following activity that you have done in the previous grade to guide you towards a solution
The crane in the port of Durban responds to six different input commands:

1. Left
2. Right
3. Up
4. Down
5. Grab
6. Release

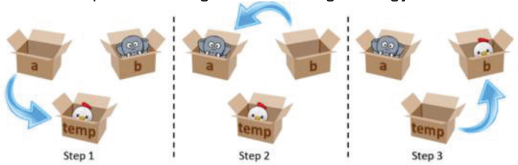


https://olympiad.org.za/talent-search/past-papers/pen-and-paper/

Crate A is in the left position, crate B is in the position on the right
Which is the correct set of instructions to swap the position of the two crates? Write down the letter of the correct answer.
A (Down, Grab, Up, Right, Down, Release, Up)
B (Down, Grab, Up, Right, Down, Release, Up) (Right, Down, Grab, Up, Left, Left. Down, Release, Up) (Right, Down, Grab, Up, Right, Down, Release)
C (Right, Right, Down, Grab, Up) (Left, Left, Down, Release, Up)
D (Down, Grab, Up, Right, Right, Down, Release, Up) (Down, Grab, Left, Down, Release, Up) (Down, Grab, Up, Right, Down, Release, Up

Note:
Learners need to understand that when swopping two variables, they need an extra variable to temporary store a variable to be swopped.

One can explain it through the following analogy

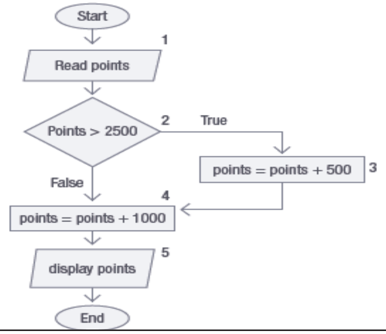


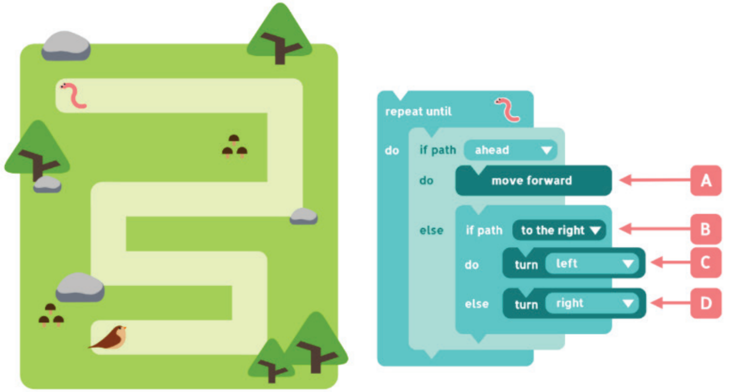
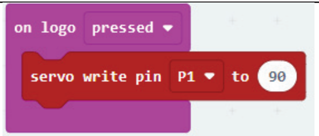
C.5 Evaluate a given solution towards potential improvement	

Example activity

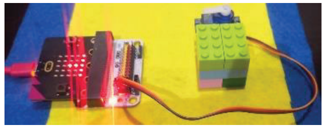
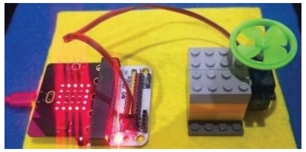
A customer is awarded points when using a loyalty card. The store decides double all its customers' points. Those customers who have points greater than 2500 before the points are doubled will be awarded an additional 500 points.
Study the flowchart and evaluate the outcome of the program by answering the questions that follow:

- If Box 4 is moved immediately below Box 1, will the algorithm still produce the same result if the points entered are 1000? Explain your answer.
- If Box 4 is moved immediately above Box 1, will the algorithm still produce the same result if the points entered is 3000? Explain your answer.
- What will happen if Box 3 and Box 4 are interchanged.



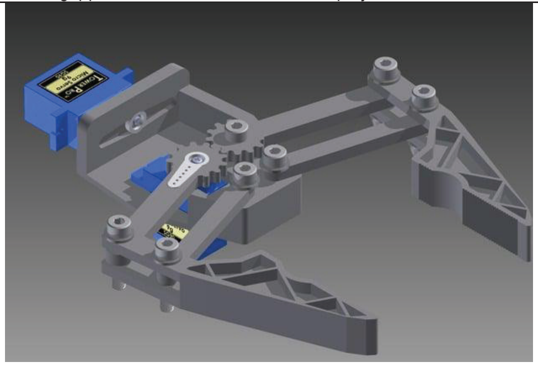
Content (Grade 8 / Term 3)	Notes/Examples
C.6 Recognise and interpret patterns in symbolic sets of data or visualisations	**Link to C.1 - C.5 and C.7 as well as R.5 – R.6**
Example activity The bird must catch the worm, but there is a mistake in the code that the bird must follow Which part of the block code contains a mistake?, A, B, C or D? [2020-TS-Intermediate-Question-Paper.pdf (olympiad.org.za)]()  repeat until do if path ahead do move forward ← A else if path to the right ← B do turn left ← C else turn right ← D	
C.7 Create or complete a pattern to represent a data set	**Link to C.6**
Example activity Refer to the activity in C.3 above. In pairs, learners design, create and code their own patterns	**Link to C.3**
Robotics	**Link to R.1 – R.6**
R.3 Outline the different components of a robot	
Revise the electronic components and sensors and actuators covered thus far. **Servo Motors** Learners must explain what the following segments of code will do when executed on a 180-degree servo as opposed to a 360-degree continuous servo, and apply it in an application as required  on button A pressed servo write pin P1 to 180  on button B pressed servo write pin P1 to 0  on logo pressed servo write pin P1 to 90 The learners should be able to explain what the following segments of code will do when executed on a 360-degree continuous servo and apply it in an application as required.  on button A pressed continuous servo P1 run at 50 %  on button B pressed stop servo P1  on logo pressed set servo P1 angle to 0 °	**Learners must** • differentiate between a 180-degree servo and a 360-degree (continuous rotation servo), in terms of its mechanical operation and how its coded. • present a basic understanding of the concept of pulse width modulation (PWM) • distinguish between a 180 degree and a 360 degree servo motor and apply both where appropriate

Content (Grade 8 / Term 3)	Notes/Examples

180 Degree Micro Servo		360-Degree Continuous Servo	
Mechanical Operation		**Mechanical Operation**	
A 180-degree micro servo is a type of servo motor that can rotate precisely within a 180-degree range. It has a limited range of motion, typically from 0 degrees (fully counterclockwise) to 180 degrees (fully clockwise). It is designed for applications where precise angular control within this limited range is required, such as steering mechanisms or robot joints that do not need continuous rotation. It operates by receiving PWM signals where the duty cycle of the PWM signal corresponds to the desired servo position. The servo's shaft position is determined by the width of the high (on) pulse in the PWM signal, with a typical range of 1 to 2 milliseconds, where 1 ms corresponds to 0 degrees, and 2 ms corresponds to 180 degrees.	 	A 360-degree continuous servo, on the other hand, can rotate continuously in either direction without any physical stops. It doesn't have a fixed endpoint like a 180-degree servo; it can rotate indefinitely. Continuous servos are commonly used in applications where continuous motion or speed control is needed, such as robot wheels or conveyor belts. It also uses PWM signals for control, but the position is not fixed; instead, the duty cycle corresponds to the speed and direction of rotation. For a continuous servo, the centre (neutral) position is usually around 1.5 milliseconds, but you can adjust this to stop the servo based on the specific servo's characteristics.	 

Also see: https://makecode.microbit.org/reference/servos and https://littlebirdelectronics.com.au/guides/191/servo-with-micro-bit

The learners need to present a conceptual understanding of how different servos could be used to create mechanical actuators and robot arms.

Outline the basic engineering principles behind how servo motors could be used to create grippers and actuators in a robotics project.	Outline the basic engineering principles behind how servo motors could be used to create simple robotic arms to mimic movement.
 https://www.thingiverse.com/thing:715525	 https://www.instructables.com/How-to-Make-Record-and-Play-Servo-Based-Robotic-Ar/

R.5 Design a simple artefact based on a set of design instructions

R.6 Mimic the operations of a robot

Example activity – Control a continuous servo and LEDs using radio communication

The direction of the servo is changed by tilting the microcontroller and the LEDs by sending signals when buttons A, B and the logo are pressed

Notes/Examples

Link to R.7

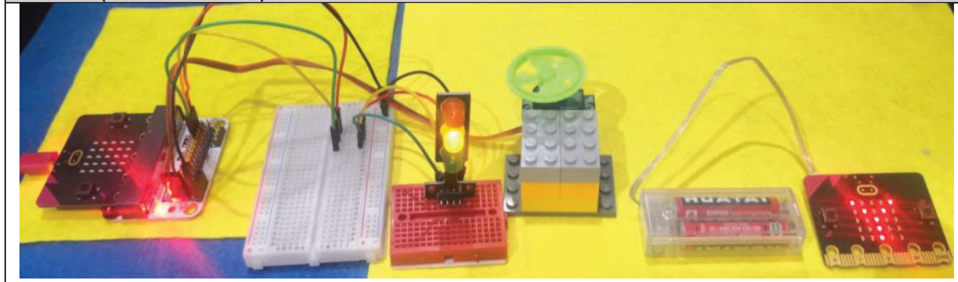
R.5 and R.6 done together

Revise the concept of communication between two devices

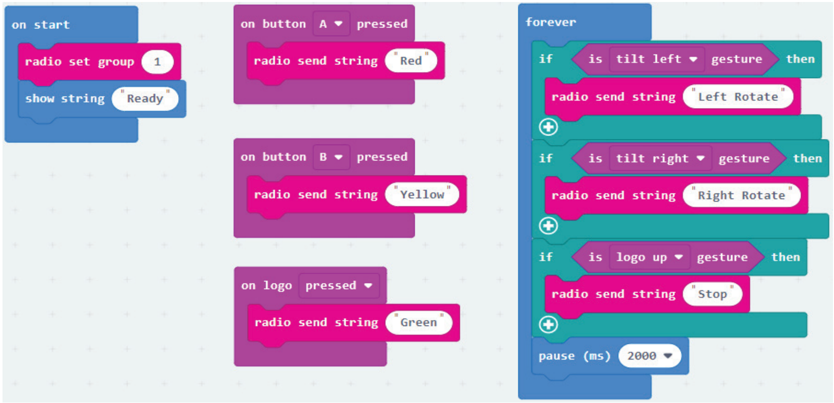
In Grade 7 simple radiocommunication between two devices was done using numbers.

In Grade 8 strings are also passed and compared.

Content (Grade 8 / Term 3)



Sender App



```
on start
    radio set group 1
    show string "Ready"

on button A ▾ pressed
    radio send string "Red"

on button B ▾ pressed
    radio send string "Yellow"

on logo pressed ▾
    radio send string "Green"

forever
    if is tilt left ▾ gesture then
        radio send string "Left Rotate"
    ⊕
    if is tilt right ▾ gesture then
        radio send string "Right Rotate"
    ⊕
    if is logo up ▾ gesture then
        radio send string "Stop"
    ⊕
    pause (ms) 2000 ▾
```

Receiver App

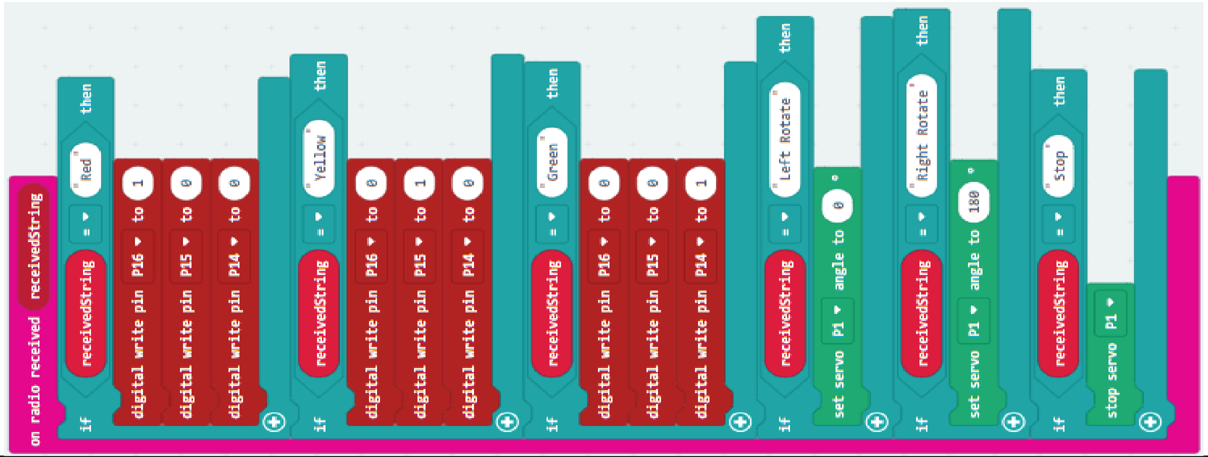
```
on button A ▾ pressed
    digital write pin P16 ▾ to 0
    digital write pin P15 ▾ to 0
    digital write pin P14 ▾ to 0
    stop servo P1 ▾

on start
    radio set group 1
    digital write pin P16 ▾ to 0
    digital write pin P15 ▾ to 0
    digital write pin P14 ▾ to 0
    show string "Ready"
    play tone Middle C for 1/8 ▾ beat until done ▾
```

Notes/Examples

Project

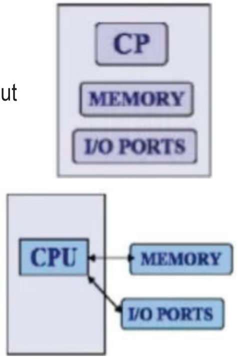
The learners must be able to apply the principles of design thinking to develop an artefact where radiocommunication are used for control purposes.

For the project, a minimum of ONE sensor or actuator should be used, as well as any number of additional hardware components (LED's Buzzers, Buttons etc)

Content (Grade 8 / Term 3)	Notes/Examples
	

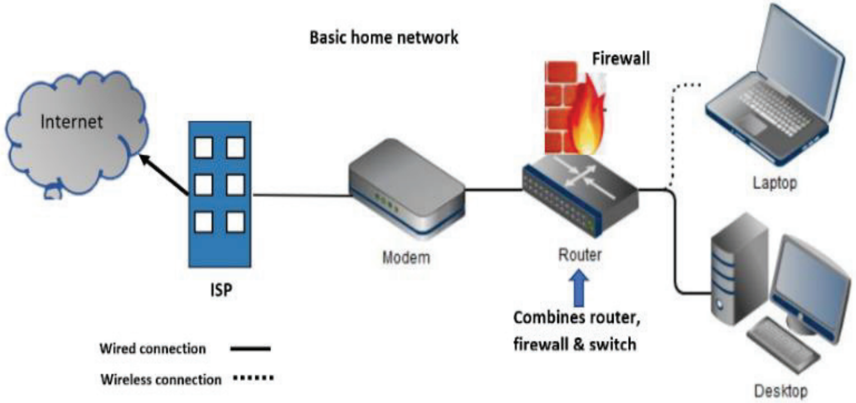
```
on radio received receivedString
  if receivedString = "Red" then
    digital write pin P16 to 1
    digital write pin P15 to 0
    digital write pin P14 to 0
  if receivedString = "Yellow" then
    digital write pin P16 to 0
    digital write pin P15 to 1
    digital write pin P14 to 0
  if receivedString = "Green" then
    digital write pin P16 to 0
    digital write pin P15 to 0
    digital write pin P14 to 1
  if receivedString = "Left Rotate" then
    set servo P1 angle to 0 °
  if receivedString = "Right Rotate" then
    set servo P1 angle to 180 °
  if receivedString = "Stop" then
    stop servo P1
```

R.7 Create, test and execute a set of robotic instructions	Link to R.5 and R.6
Example activity - Pet feeder The purpose of the game is to make the size of a pet grow or shrink depending on whether the pet is fed or not. • Start with the smallest circle or smallest square on the microcontroller (your pet). Calculate the area for the initial circle or square. (Assume radius=1 or length=1). • When you press button A, the pet must "grow" with 1 unit. Calculate and display the new area and display an image of the larger pet. • When you press button B, the pet must "shrink" with 1 unit. Calculate and display the new area and an image of the smaller pet. • If the pet grows beyond the boundaries of the microcontroller, display an "explode" icon. • If the pet shrinks so that it is smaller than its initial size, display a "disappear" icon.	
Digital Concepts	
D.1 Outline the concept of technology and purpose of information technology (IT)	**Link to D.1, D.3, D.4, D.5 and D.7,**
D.2 Recognise that he or she is living as citizens in a digital world.	
Present an understanding of the dangers of the online environment. Explain online safety, responsible use of social media (WhatsApp/TikTok/Instagram etc), POPI Act. **Example activity 1**: Define responsible use of social media and its various forms. Discuss the emotional and psychological impact of social media on individuals. Share examples of social media incidents. **Example activity 2**: 1. Can you come up with a real-life scenario where online safety measures would be crucial to prevent harm or protect personal information? 2. Share an example of a responsible social media post and explain why it demonstrates responsible use. 3. Discuss the potential consequences of sharing personal information online without considering privacy settings.	**Link to D.6**
D.3 Demonstrate an understanding of the concept of a computing device.	**Link to D.3, D.4 and D.6 and R.3**
Example activity – Microprocessors vs Microcontrollers Today, both microprocessors and microcontrollers are two essential components of modern electronics.	Introduce the cocept of microprocessors or microcontrollers as computing devices Learners need to, at a basic level, • Understand what a microprocessor is

<table>
<tr><th>Content (Grade 8 / Term 3)</th><th>Notes/Examples</th></tr>
<tr><td>
Content: Definition and Functions:
Microcontroller
• A microcontroller is a small, integrated computer that is designed for specific tasks, applications or embedded systems.
• It includes a relatively simple CPU (Central Processing Unit), combined with peripheral devices such as memory, input/output ports, and timers on a single chip.
Microprocessor
• A microprocessor is a general-purpose computer component that executes instructions from a computer program and a general-purpose computing device. It can perform various functions depending on the program
• The microprocessor chip does not typically have built-in memory or input/output capabilities like a microcontroller.


Differences:
1. microcontrollers are designed for specific tasks or embedded systems, while microprocessors are the central processing units (CPUs) in computers used in multi-purpose computing devices.
2. Create a table on the board or a slide (if using a projector) to compare microcontrollers and microprocessors:
<table>
<tr><th>Aspect</th><th>Microcontroller</th><th>Microprocessor (CPU)</th></tr>
<tr><td>Purpose</td><td>Specific tasks or functions</td><td>General-purpose computing</td></tr>
<tr><td>Input/Output ports</td><td>Yes (on chip)</td><td>No (separate)</td></tr>
<tr><td>Memory</td><td>Included (on chip)</td><td>Separate</td></tr>
<tr><td>Examples</td><td>Arduino, Microcontroller,</td><td>Intel, AMD processors</td></tr>
</table>
Applications:
• Discuss some common applications of microcontrollers:
    • Microcontrollers are often used in devices like washing machines, microwave ovens, remote controls, and embedded systems.
    • They are excellent for tasks that require real-time control and low power consumption.
    • The microcontroller is an example of a microcontroller as it is a small device that can run programs and interact with sensors, buttons, LEDs and other components
• Discuss common applications of microprocessors:
    • Microprocessors power personal computers, laptops, smartphones, and servers.
    • They are suitable for tasks that require high computational power and multitasking capabilities.
</td><td>
• Understand what a microcontroller is
• Provide an example of each
• Differentiate between a microcontroller and a microprocessor

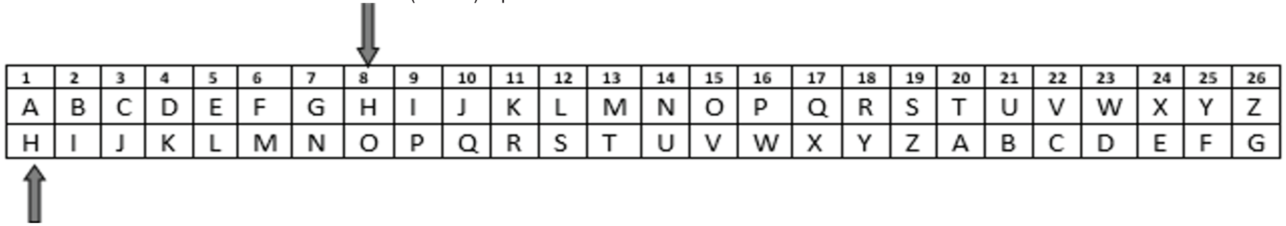
Example:
Microcontroller is a small computing device that can run programs and interact with sensors, buttons, LEDs and other components
</td></tr>
<tr><th>D.4 Identify the common uses of ICT in the real world</th><th>Link to D.5 and D.6</th></tr>
<tr><th>D.5. Differentiate between the components of an ICT system</th><th>Link to D.5</th></tr>
<tr><td>
Introduce the following concepts firewall in a network,
Internet service provider (ISP), internet address.

Use a firewall to protect network resources from outside intruders


</td><td>
D.3, D.4 and D.5 are done together

Revise
• Networks
• Network devices
• Connections (wired and wireless)
Introduce concept of Firewall, Internet Service Provider (ISP), internet address
</td></tr>
</table>

<table>
<tr><th>Content (Grade 8 / Term 3)</th><th>Notes/Examples</th></tr>
<tr><td>D.6 Explain how the adaptation of technology impacted the world we work and live in</td><td>Done in relation to C.1 and C.2 and C.3</td></tr>
<tr><td>Discuss the social and global impact of networks, including their role in communication, society.
<u>Example activity</u>:
How would networks contribute to disaster response?
Discuss the use of networks in transportation and logistics for tracking vehicles, shipments and improving efficiency.
Discuss how networks are employed in education and online learning platforms.</td><td>Learners must understand the role of networks and how they impact the world we live in and provide examples to illustrate the impact.</td></tr>
<tr><td>D.8 Interpret a pattern to represent or communicate a message or image.</td><td>D.8 and D.9 done together</td></tr>
<tr><td>D.9 Create a pattern to represent or communicate a message or image.</td><td rowspan="2">D.8 and D.9 done together

Cipher - a method of hiding words or text with encryption by replacing original letters with other letters, numbers and symbols through substitution or transposition. A combination of substitution and transposition is also often employed.
It is therefore any method of transforming a message to conceal its meaning.
Encryption is to convert information or data into secret code
Decryption is to convert a secret (<i>encrypted</i>) message to an understandable (<i>decrypted</i>) message</td></tr>
<tr><td>Example activity – decrypt and encrypt a message using a cipher
A Caesar cipher (named after the Roman Emperor Julius Caesar) is a simple encryption method in which each letter in a word is replaced by another letter in the alphabet, depending on the ‘shift’ key.
The alphabet can be shifted up to 25 places but shifting it 26 places takes it back to its original position and shifting it 27 places is the same as shifting it 1 place.
For example, if the ‘shift’ key is 7, the letter A will be replaced by the letter H (as shifting 7 positions from position 1 takes one to the 8th letter in the alphabet (H) – H therefore becomes the 1st letter of the ‘new’ (shifted) alphabet as shown below:
<table><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td></tr><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td><td>G</td><td>H</td><td>I</td><td>J</td><td>K</td><td>L</td><td>M</td><td>N</td><td>O</td><td>P</td><td>Q</td><td>R</td><td>S</td><td>T</td><td>U</td><td>V</td><td>W</td><td>X</td><td>Y</td><td>Z</td></tr><tr><td>H</td><td>I</td><td>J</td><td>K</td><td>L</td><td>M</td><td>N</td><td>O</td><td>P</td><td>Q</td><td>R</td><td>S</td><td>T</td><td>U</td><td>V</td><td>W</td><td>X</td><td>Y</td><td>Z</td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td><td>G</td></tr></table>Therefore, the encryption for the word ‘HELLO’, will become ‘OLSSV’.
Do the following:
i) Use the above information to encrypt the following message: EIGHT NOON AT CAGE
ii) Decrypt– the opposite of encrypt (i.e. crack the secret message) the following secret message WHZZDVYK PZ JLHZHY
iii) Use your own key to encrypt the following message:
iv) Give your encrypted message with the key to a friend and ask your friend to decrypt the message.</td></tr>
<tr><td>D.10 Demonstrate a basic proficiency in the application of digital skills.</td><td>Link to R.5</td></tr>
<tr><td>Learners explore presentation software, e.g. PowerPoint to be used for reporting back after pair or group activities or to present coding solutions, robotics projects, etc.</td><td></td></tr>
</table>

3.2.4 Term 4

Content (Grade 8 / Term 4)	Notes/Examples
Coding	
C.1 Apply computational thinking (CT) skills to develop a set of logical instructions to solve a problem.	**Link to C.2 – C.7 and R.5 – R.7**
C.2 Present a simple coding solution using symbolic or written statements representing sequences of commands, single repetition, and conditional constructs.	**LC.1 and C.2 and C.4 are done together**

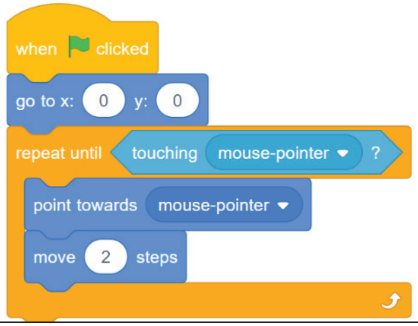
Example activity – Cat and mouse game

In pairs, use your computational thinking skills to develop your own game like the cat and mouse game described below:

The cat is chasing the mouse and tries to capture it. It involves constant pursuit, near captures, and repeated escapes. The "cat" is unable to secure a definitive victory over the "mouse", who, despite not being able to defeat the cat, is able to avoid capture.

Step 1: Cat

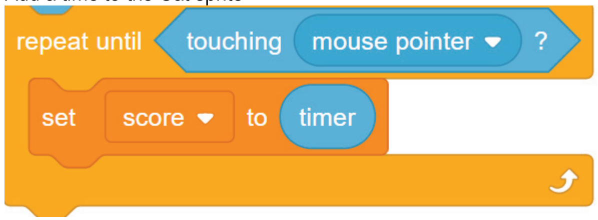
Use Cat sprite and add code for Cat sprite to continue moving until the mouse pointer is reached



```
when [flag] clicked
go to x: 0 y: 0
repeat until <touching mouse-pointer ▾ ?>
    point towards mouse-pointer ▾
    move 2 steps
```

Step 2:

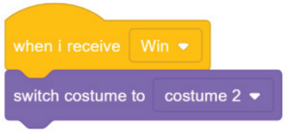
Add a time to the Cat sprite



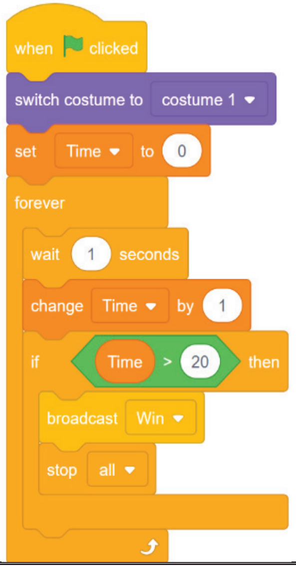
```
repeat until <touching mouse pointer ▾ ?>
    set score ▾ to timer
```

Step 3: Add another sprite

Make an empty sprite with two costumes: one blank (costume 1) and one with a "Win" banner (costume 2). Add code for this sprite to make a variable timer:



```
when i receive Win ▾
switch costume to costume 2 ▾
```



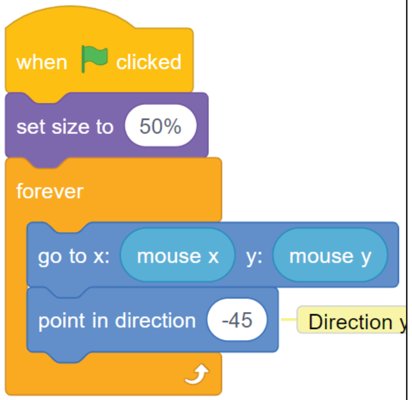
```
when [flag] clicked
switch costume to costume 1 ▾
set Time ▾ to 0
forever
    wait 1 seconds
    change Time ▾ by 1
    if <Time > 20> then
        broadcast Win ▾
        stop all ▾
```

Step 4: Add the mouse sprite

Add the mouse sprite and code for the mouse sprite

For the main cat sprite, change the "touching" and "point" to "Mouse" instead of "Mouse-Pointer"

CAT AND MOUSE - SCRATCH WIKI (SCRATCH-WIKI.INFO)



```
when [flag] clicked
set size to 50%
forever
    go to x: mouse x y: mouse y
    point in direction -45
```

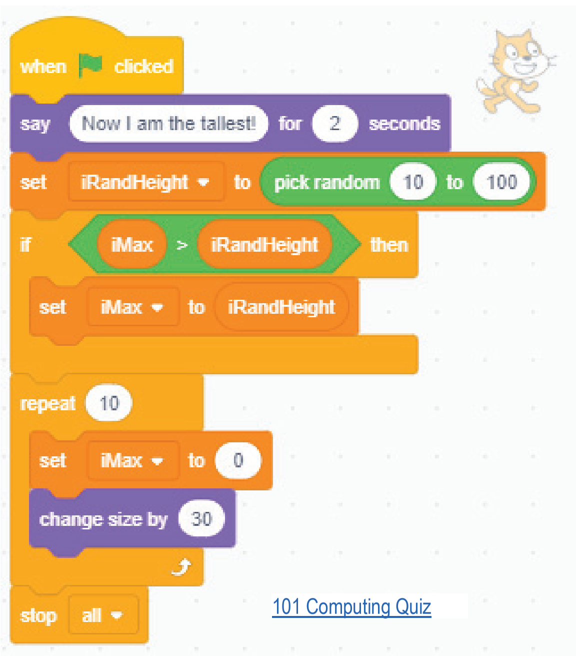
Note

Learners study the code of the cat and mouse game, use computational thinking and their experience from studying the code and what they have learned so far in terms of coding to develop their own, similar game.

Remember

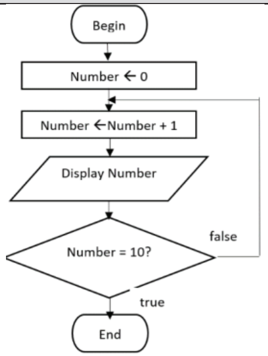
It is important that learners plan a program. If code is not planned well before writing it, it can lead to difficulties when they need combining structures, as they may not have a clear roadmap for how different structures should work together.

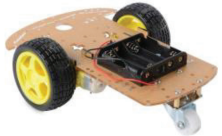
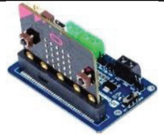
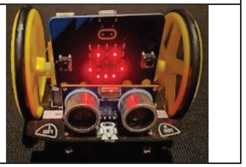
Content (Grade 8 / Term 4)	Notes/Examples
C.3 Interpret and execute a given symbolic or written set of commands	
Example activity 1 – Code a written set of commands for a number guessing game Wite code to implement the following game: You think of a number from 1 to 20. Your classmate must guess which number you are thinking of and tell you. You may only respond with 'lower' (if your classmate must guess a lower number) or 'higher' (if your classmate must guess a higher number). Your classmate has only five guesses to guess the right number. Your classmate keeps guessing and you keep responding until the number guessed is correct or until the five guesses have been used up. Write a computer program to simulate the above game using the algorithm below: • Generate a random number between 1 and 20. • Ask the user to guess and enter the number 'you are thinking of' (the randomly generated number). • Test if the number the user entered is smaller or larger than the random number or equal to the random number. • If the user guessed too high, tell the user that the next guess must be lower. • If the user guessed too low, tell the user that the next guess must be higher. • If the user guessed the correct number, congratulate the user for guessing the correct number. • The program must execute until the user guesses the correct number or until the user had five guesses. • If the user guessed the number correctly, display the number of attempts that the user tried to guess the number. • If the user didn't guess the number correctly, display the number that the user had to guess.	
C.4 Debug a given symbolic or written set of instructions	**Link to C.1, C.2**
Done with C.1 and C.2	
C.5 Evaluate a given solution towards potential improvement	

Example activity – Evaluate and improve code
Study the following problem statement, then rearrange the code provided to create a working program for the problem statement:
Problem statement:
The Scratch program needs to determine who the tallest learner in a class of an unknown number of learners is. Each learner is represented by a number, e.g., the first learner is 1 and the last learner in the class is **n**, where **n** represents the size of the class. The code supplied is incorrect. There are some logical errors in the code and not all blocks are used to obtain the correct result.
The following represents a high-level algorithm for the problem (you need to refine the algorithm and implement the algorithm in Scratch)
• A random number between 100 and 200 (cm) must be generated each time to represent the height of the n learners.
• For each learner's height generated, check if the learner is taller than the previous learner.
• If the learner is taller than the previous learner, then
  ○ The variable storing the tallest learner's height will change.
  ○ A message will be displayed: Learner no? is now the tallest".
• When the height of all learners was checked, display a message:" Learner no? is the tallest learner in class"

101 Computing Quiz
	You want to interleave practise in problems.
	"It is important that problem types must differ, for example, you want to randomly have a problem of one type and then solve a problem of another type and then a problem of another type. And in doing that, it feels difficult, and it doesn't feel fluent. And the signals to your brain are, I am not getting this. I am not doing very well. But in fact, that effort to try to figure out what kinds of approaches do I need for each problem as I encounter a different kind of problem, that is producing learning. That is producing robust skills that stick with you."
	Dr Mark A. McDaniel, Harvard University

```
when [flag] clicked
say [Now I am the tallest!] for (2) seconds
set [iRandHeight ▾] to (pick random (10) to (100))
if <(iMax) > (iRandHeight)> then
    set [iMax ▾] to (iRandHeight)

repeat (10)
    set [iMax ▾] to (0)
    change size by (30)

stop [all ▾]
```

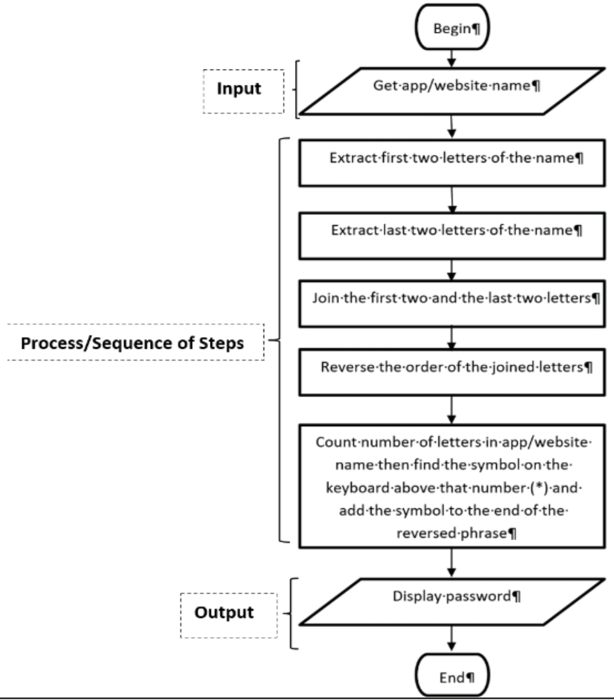
<table>
<tr><th>Content (Grade 8 / Term 4)</th><th>Notes/Examples</th></tr>
<tr><td>C.6 Recognise and interpret patterns in symbolic sets of data or visualisations.</td><td>Link to C.1 – C.5 and C.6 and R.6</td></tr>
<tr><td>
Example activity
Study the flowchart on the right and answer the questions that follow:
a. What will be displayed?
b. Explain in your own words what the purpose of the algorithm is.
c. How many times is the instruction
Number ← Number +1
executed?
d. If the instruction
Number ← 0
is changed to
Number ←- 1
a. what will then be displayed?
b. how many times will the loop then be executed?



</td><td></td></tr>
<tr><td>C.7 Create or complete a pattern to represent a data set</td><td>Link to C.1 – C.6 and R.6</td></tr>
<tr><td>Robotics</td><td></td></tr>
<tr><td>R.1 Explain what a robot is in simple terms.</td><td>Link to R.2</td></tr>
<tr><td>R.2 Identify different types of robots.</td><td>Link to R.1</td></tr>
<tr><td>R.3 Outline the different components of a robot</td><td></td></tr>
<tr><td>R.5 Design a simple artefact based on a set of design instructions</td><td></td></tr>
<tr><td>R.6 Mimic the operations of a robot</td><td>Link to C.7, R.5 and R.7</td></tr>
<tr><td>R.7 Create, test and execute a set of robotic instructions</td><td>Link to R.5 and R.6</td></tr>
<tr><td>
Example project – obstacle avoidance robot
Learners investigate and describe the typical composition of an obstacle avoidance robot using an ultrasonic sensor, applying the principles of design thinking.
The learners should be able to describe the use and purpose of the following components typically used as part of an obstacle avoidance robot (on a conceptual level only)
- Chassis
- Wheels
- Power supply
- Microcontroller
- Motors and
- Motor driver (On a conceptual level, more detail in grade 9)
- Breadboard and jumper wires for prototyping
- Ultrasonic sensor
The learners should also be able to describe the logical principles guiding the code of an OAR.
IMPORTANT The leaners do not need to construct the OAR (Obstacle avoidance robot). The teacher may demonstrate the operation of such a robot, based on a prebuild kit. Schools which the facilities may decide to construct the robot and potentially demonstrate the 3-D printing process for enrichment purposes.
<table>
<tr><th>Example of a chassis with wheels and motors</th><th>Empty battery back</th><th>DC Motor with wheel</th><th>Continuous servo</th><th>Microcontroller and motor driver</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
</table>
Nice resource link: https://www.robotique.tech/robotics/obstacle-detection-system-with-microbit/
(Also see: https://www.instructables.com/DIY-Microbit-Educational-Mobile-Robot-V2/ - Dame, A DIY Microcontroller Educational Mobile Robot V2)
</td><td>
Note:
The inclusion of the OAR is to introduce the learners to the practical principles behind the operation of the robot. It is not expected of the learners to construct the robot, only to describe its basic principles of operation, the composition of such a robot and the code logic behind it. Schools that do have the resources could include the building, as an enrichment activity in groups.
Obstacle avoidance robot (OAR) using an ultrasonic sensor.
<table>
<tr><th>Ultrasonic sensor</th><th>Example of an OAR using an ultrasonic sensor</th></tr>
<tr><td></td><td></td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Content (Grade 8 / Term 4)</th><th>Notes/Examples</th></tr>
<tr><td>Digital Concepts</td><td></td></tr>
<tr><td>D.1 Outline the concept of technology and purpose of information technology (IT)</td><td>D.1, D.3, D.4, D.5, D.7 done together</td></tr>
<tr><td>Case study: Exploring online gaming
Learners work in pairs.
The purpose of this case study is to help learners understand the essential concepts of computer networks, with a specific focus on how these networks impact online gaming experiences through input, processing, and output.
Instructions: Imagine you and your friends are online gamers. You want to optimise your online gaming experience and minimise lag while playing multiplayer games. To do this, you decide to explore how computer networks handle data during online gaming sessions.
Prepare a short report summarising your understanding of how computer networks affect online gaming.
Address the role of input devices, network setup, data processing, and server interaction in online gaming.
Make a list of devices needed and a short description of why this is necessary.
Draw a possible network setup showing connections and layout.
<table>
<tr><th>Input</th><th>Processing</th><th>Output</th></tr>
<tr><td></td><td></td><td></td></tr>
</table></td><td></td></tr>
<tr><td>D.2 Recognise that he or she is living as citizens in a digital world.</td><td>Done with to D.6</td></tr>
<tr><td>Case Study: Protecting Personal Data in a Public Wi-Fi Scenario
Learners work in pairs and watch the following video: https://youtu.be/X87MJEsJ91A and https://youtu.be/bdVkkRmJEeM then discuss the following:
Sarah, a freelance graphic designer, often finds herself working remotely from coffee shops and public places to escape the confines of her home office. She relies on public Wi-Fi networks to connect to the internet and communicate with clients. One day, while working at a popular coffee shop, she encounters a network security and privacy challenge. Sarah values the convenience of working from various locations, but she's aware of the potential security risks associated with public Wi-Fi networks. She understands that her sensitive client data and personal information are at risk if not adequately protected. While sipping her coffee and working on a client's project, Sarah notices that the public Wi-Fi network at the coffee shop is unsecured. She overhears another person discussing the same concern with a friend - that anyone on the same network could potentially intercept their data. Sarah realises that she might not be taking adequate precautions to protect her privacy and sensitive client information.

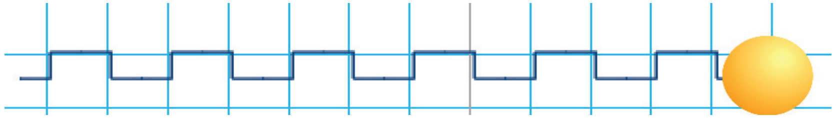
Privacy Concerns: What specific privacy risks does Sarah face while working on an unsecured public Wi-Fi network?
Data Security: What are the potential consequences of her client data being intercepted or compromised?
Safe Practices: What steps can Sarah take to enhance her network safety and privacy while working in public places?
Educating Others: How can Sarah educate fellow coffee shop customers about network security and privacy concerns?
Legal and Ethical Considerations: What legal and ethical responsibilities does Sarah have regarding the protection of her client's data?
Client Communication: How should Sarah communicate with her clients about the steps she takes to safeguard their information when working remotely?</td><td>Learners need to understand
• what data security and data privacy and data protection are and why it is important
• The role of POPI act</td></tr>
<tr><td>D.3 Demonstrate an understanding of the concept of a computing device.</td><td>D.1, D.3, D.4, D.5, D.7 done together</td></tr>
<tr><td>D.4 Identify the common uses of ICT in the real world</td><td></td></tr>
<tr><td>Imagine you are tasked with helping a family set up a secure home network. The family wants to connect multiple devices such as computers, smartphones, and smart appliances while ensuring their network is safe from potential threats.
Discuss the steps taken to set up and secure the home network, including any adjustments made based on the family member's concerns, e.g. security, space, etc.
Learners will also participate in a group discussion, sharing their reflections on the importance of network security in the context of home networks.</td><td></td></tr>
<tr><td>D.5 Differentiate between the components of an ICT system</td><td>D.1, D.3, D.4, D.5, D.7 done together</td></tr>
<tr><td>D.6 Explain how the adaptation of technology impacted the world we work and live in</td><td>Done with to D.2.</td></tr>
<tr><td>D.7 Present a basic understanding of the concept of input processing and output.</td><td>D.1, D.3, D.4, D.5, D.7 done together</td></tr>
<tr><td>D.8 Interpret a pattern to represent or communicate a message or image</td><td>Link to D.9 and C.2 – C.5 and R.5 – R.7</td></tr>
</table>

Content (Grade 8 / Term 4)	**Notes/Examples**
Example activity: Communicate/represent an algorithm using steps of how to solve a problem. Create steps to solve the following problem. Start by taking a word, e.g. “Telegram”. Create a password by swapping the first 2 letters and the last 2 letters. Reverse the order of the joined letters and remove the remaining letters. Count the number of letters and add the number at the end as symbol on the keyboard, e.g. 8 will become *. Telegram will become meaT*. **Steps** 1. Take the first two letters of the app or website, e.g. Te for Telegram 2. Take the last two letters of the app or website, e.g. am for Telegram 3. Join the first two and the last two letters, e.g. Team 4. Reverse the order of the joined letters, e.g. meaT 5. Now, count the number of letters in the app or website name, e.g. Telegram = 8 letters, then find the symbol on the keyboard above that number (*) and add the symbol to the end of the reversed phrase, e.g. meaT*. 	
D.9 Create a pattern to represent or communicate a message or image	**Done with D.8.**
D.10 Demonstrate a basic proficiency in the application of digital skills.	**Link to Cs and Rs**
Continuously combine application skills with Cs.	

3.3 GRADE 9

Note:
Teachers must include the following competencies and content in their Annual Teaching Plan (ATP), distributed across the terms and sequenced, organised and grouped in a manner that will facilitate learning in a manner that will make sense for learning and teaching, maximize the learners' learning outcomes and achievement. and in a way that will make optimal use of time and resources. Some competencies could also be combined in bigger/more complex activities/programs.

3.3.1 Term 1

Content (Grade 9 / Term 1)	Notes/Examples
Coding	
C.1 Apply computational thinking (CT) skills to develop a set of logical instructions to solve a problem.	**Link to C.2 – C.7, R.5 – R.7**
C.2 Present a simple coding solution using symbolic or written statements representing sequences of commands, single repetition, and conditional constructs.	**Link to C..1 and C.3 – C.7 and D.8**
Example activity Use computational thinking to develop instructions (algorithm) for drawing the following pattern:  1) **Focus on the main steps –** identify the main steps (develop a high-level algorithm) – abstraction The ball moves forward, up, and down 2) Refine – **break the main steps down** into small, logical steps (decomposition) **Movement:** Forward, up, forward, down forward, up, forward down, forward, up forward down, etc. FW (15 steps) up (15 steps) FW (30 steps), down (15 steps), Forward 30 steps, etc. ***One could look at 30 steps as 15 steps + 15 steps*** to enhance the pattern, e.g. **Movement:** FW 15, UP 15, FW 15, FW 15, DOWN 15, FW 15, FW 15, UP 15, FW 15, FW 15, DOWN 15, FW 15, FW 15, UP 15, FW 15, FW 15, DOWN 15, FW 15……. FW 15, UP 15, FW 15, FW 15, DOWN 15, FW 15 **3) Find the pattern** FW 15, UP 15, FW 15, FW 15, DOWN 15, FW 15 – repeated 6x  To go up – turn left; to move turn right To go down – turn right; to move, turn left 4) Use the steps and pattern to **develop the algorithm** that solves the problem Face east, FW 15, turn left, FW 15, turn right FW 15, FW 15, turn right, forward 15, turn left FW 15 – repeated 6x East = direction 90°; UP = direction 0°; DOWN = direction 180° 5) **Code the algorithm** (convert to block-based code) Repeat 6 Direction 90° Move 15 steps Direction 0° Move 15 steps Direction 90° Move 15 Steps Move 15 Steps Direction 180°	**C.1 and C.2 done together** **Learners need to understand:** Recognising patterns helps one to identify similarities, shared characteristics and/or differences within or between problems. This helps to make predictions or to generalise solutions. If one can identify a pattern, one is generally able to predict, for example, the next item in a series or the next step in completing a task. **Pattern** – The way in which something is arranged (usually with repetition) or in which it happens or is done. **Pattern recognition** is therefore the process of recognising patterns, e.g. in data or behaviour. **Pattern recognition** is based on the five key steps: • Identifying **common elements** in problems or tasks • Identifying and interpreting **common differences** in problems or tasks • Identifying **individual elements** within problems or tasks • **Describing patterns** that have been identified • Making **predictions** based on **identified** patterns (**Predictions** are the connecting links between what you already know, with new information or knowledge). **How are patterns and pattern recognition helpful?** • Identifying patterns can point you to an existing solution that you can use to solve a new problem. • Patterns help us to make **predictions** • Pattern recognition helps us to **generalise** (To **generalise** is to look at specific cases; identify a pattern or relationship that will always be true; then represent the pattern in symbolic format).

Content (Grade 9 / Term 1)	Notes/Examples
Move 15 Steps Direction 90° Move 15 steps 6) **Test and debug** the algorithm ○ Run the program ○ If the above pattern is not displayed, troubleshoot and fix	

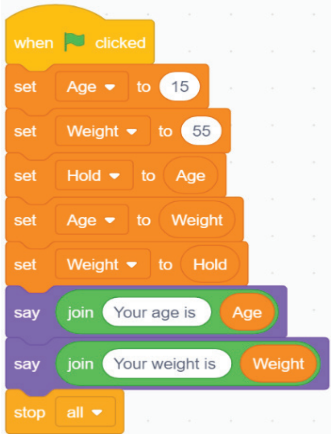
C.3 Interpret and execute a given symbolic or written set of commands

Example activity 1

Study the code on the right:

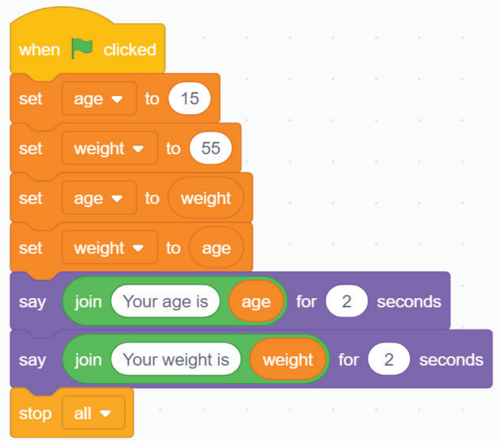
- Work through the first program in the table and provide the output.
- Explain what the first block of code does.
- Work through the second Scratch algorithm in the table and provide the output.
- Explain what the second block of code does.
- Explain the difference between the two code blocks.

First block



```
when ⚑ clicked
set Age to 15
set Weight to 55
set Hold to Age
set Age to Weight
set Weight to Hold
say join Your age is Age
say join Your weight is Weight
stop all
```

Second block



```
when ⚑ clicked
set age to 15
set weight to 55
set age to weight
set weight to age
say join Your age is age for 2 seconds
say join Your weight is weight for 2 seconds
stop all
```

Notes/Examples:

Note:

Provide learners with activities to

- read code and explain what it does.
- work through (trace) / act out code (physically or simulated) to determine the purpose/output or the correctness.

While learners should be able to describe what each line (block) of code does, (describing a code segment line-by-line) it is very important that learners explain the overall purpose of the code (program), i.e. what the program does/the purpose of the program (not a line-by-line description).

C.4 Debug a given symbolic or written set of instructions

Example activity – Get information from an ID number

An ID number contains 13 digits, e.g. 7901035123082

- The first six digits are a person's birthday, e.g. 790103
- The next four digits indicates a person's gender, e.g. 5123 If these four numbers are 5000 or bigger, the person is a male, if these digits are less than 5000, the person is a female

The program on the right asks for a person's ID and determine

- whether the person is a male of female
- provides the person's birthday, e.g. 03 January

The program does not work.

Use a trace table, find the error and correct the code and test the code.

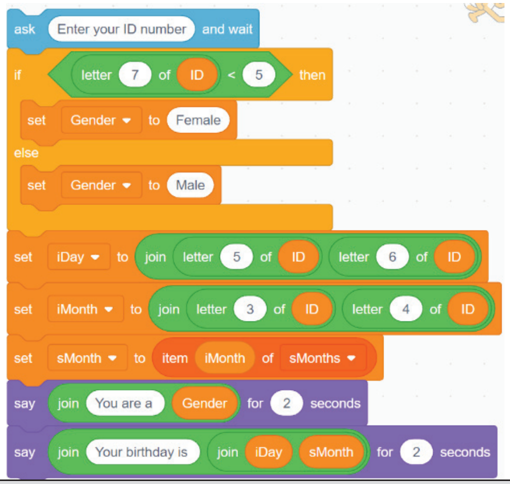
Note:

The months are stored in list sMonths

sMonths

1	January
2	February
3	March
4	April
5	May
6	June

length 12



```
ask Enter your ID number and wait
if letter 7 of ID < 5 then
    set Gender to Female
else
    set Gender to Male
set iDay to join letter 5 of ID letter 6 of ID
set iMonth to join letter 3 of ID letter 4 of ID
set sMonth to item iMonth of sMonths
say join You are a Gender for 2 seconds
say join Your birthday is join iDay sMonth for 2 seconds
```

C.5 Evaluate a given solution towards potential improvement

<table>
<tr><th>Content (Grade 9 / Term 1)</th><th>Notes/Examples</th></tr>
<tr><td>

What makes a good algorithm?

Some algorithms are better than others. So, what should we consider when we design algorithms that will ensure they are good algorithms, especially if the algorithm must be understood by non-humans such as a computer or a robot?

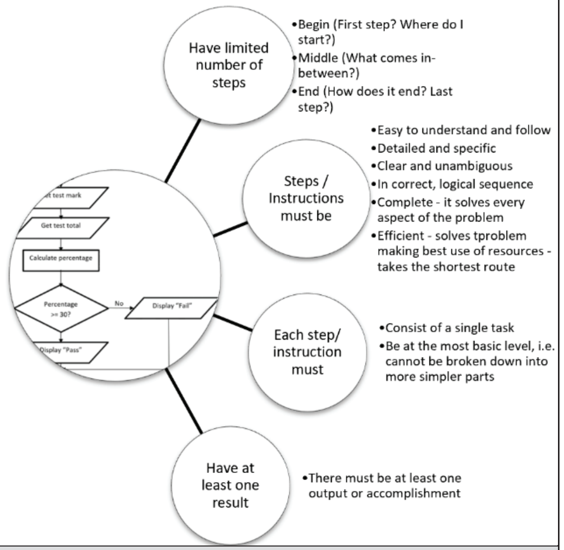
Example activity

Teacher provides examples of good and bad algorithms

Learners discuss what makes a good algorithm

</td><td>

Note: Characteristics of a good algorithm



</td></tr>
<tr><th>C.6 Recognise and interpret patterns in symbolic sets of data or visualisations.</th><th></th></tr>
<tr><td>

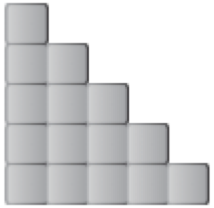
Example activity – Interpret code patterns

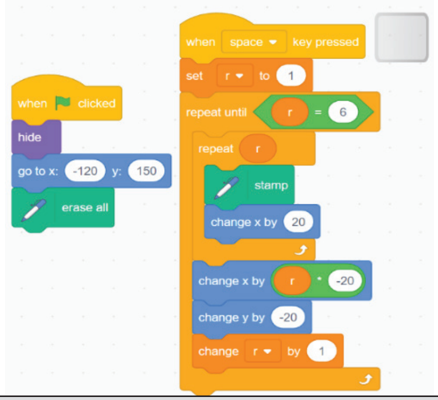
The code on the right creates uses a 'block' sprite and creates a pattern,

Rewrite and study the code.

Play around with the value of 'r' in the Repeat…Until loop and watch how the pattern changes.

Each time the value of r changes, describe and explain the 'new' pattern





```
when [flag] clicked
hide
go to x: -120 y: 150
erase all
```

```
when space key pressed
set r to 1
repeat until r = 6
    repeat r
        stamp
        change x by 20
    change x by r * -20
    change y by -20
    change r by 1
```

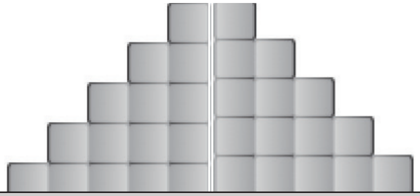
</td><td></td></tr>
<tr><th>C.7 Create or complete a pattern to represent a data set</th><th></th></tr>
<tr><td>

Example activity – Design, Create and Code patterns

Refer to the activity in C.6

Change the code to create the following pattern

See how many different patterns you can create with the 'block'

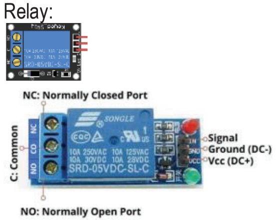
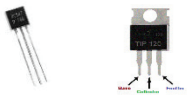


</td><td></td></tr>
<tr><th>Robotics</th><th></th></tr>
</table>

Content (Grade 9 / Term 1)	Notes/Examples
R.1 Explain what a robot is in simple terms.	**Link to R.2 and R.3**
R.2 Identify different types of robots.	**R1 and R.2 done together**
Example activity Learners investigate a robot (or drone) to rescue and explore dangerous environments, then outline the main components of the selected robot and describe how the robot is used to achieve its goals. Learners create and present a diagrammatical outline of the robot and its various components and extension capabilities. Finally, learners create and present a short report on how robots or that are used for rescue and or exploring dangerous environments could be used or deployed in space exploration.	Learners can outline different types of robots in terms of their use, composition, and application, i.e., • Artificial Intelligence (AI) Robots • Exoskeleton Robots • Industrial Robots (in relation to automation and virtual robots used in the information systems space) • AI-Powered Virtual Assistants Virtual IT Support Bots
R.3 Outline the different components of a robot	

Example activity – Introduce the learners to the concepts of switches, transistors, and relays.

The learners must present an overview of the operation of the following as part of physical computing artefacts.

Component	Basic definition	Purpose	Practical application example – May be used to:
Relay: 	An electromagnetic switch that controls the flow of current in an electrical circuit by opening or closing its contacts in response to an external signal.	Used to electrically isolate and control high-power or high-voltage devices or circuits with a low-power signal.	In robotics, relays can be used to control larger actuators or devices, such as switching power to high-torque motors or activating heavy-duty electrical equipment like pumps or solenoids.
Transistors (NPN) 	A type of bipolar junction transistor (BJT) with three layers of semiconductor material.	Used as amplifiers, switches, or signal modulators to control electrical current in various robot circuits.	Acts as switches or amplifiers in motor control, sensors, and signal processing circuits.
Additional components.			
LDR (Light-Dependent Resistor) 	A sensor whose resistance changes with variations in light intensity.	Employed to measure ambient light levels or detect changes in light conditions for various applications in robotics.	Used for light-sensitive applications like automatic lighting control or sun tracking in solar-powered robots
DC Water Pump 	A DC-powered pump designed to move water or other fluids.	Used in robotics for tasks like fluid circulation or water-based propulsion.	Facilitates fluid transport, such as water circulation in hydroponic or aquatic robots.
	A 9V/12V brushless fan is an electric device with a brushless DC motor used to circulate air and manage heat efficiently.	It serves to prevent overheating of electronic components, ensuring reliable and extended operation in robotics by actively dissipating excess heat.	Used for cooling and to allow and promote air flow in projects, for example smart biltong driers.

Introduce concepts of switches, transistors and relays

Learners must be able to present an overview
- Relays
- Transistors
- LDR (light dependent resistor)
- DC water pump
- Brushless fan

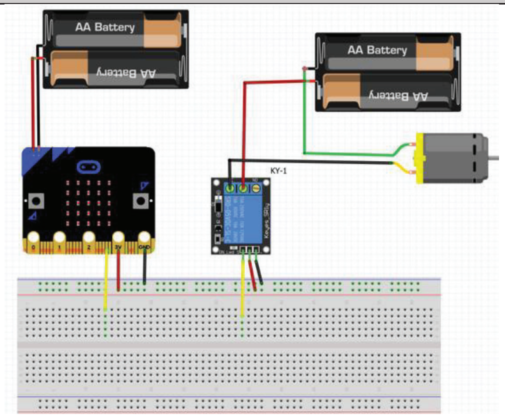
Learners must be able to
- explain why relays are used in physical computing and robotics projects. Reference should be made to the voltage and current requirements of certain components such as DC motors, (DC water pumps, etc)
- differentiate between a normally open and normally closed relay.
- present a basic understanding to the use of a transistor (NPN) in relation to acting as a switch. A TIP120 transistor is ideal for this purpose.

Nice resource link:
https://www.robotique.tech/robotics/control-a-dc-motor-with-microbit/

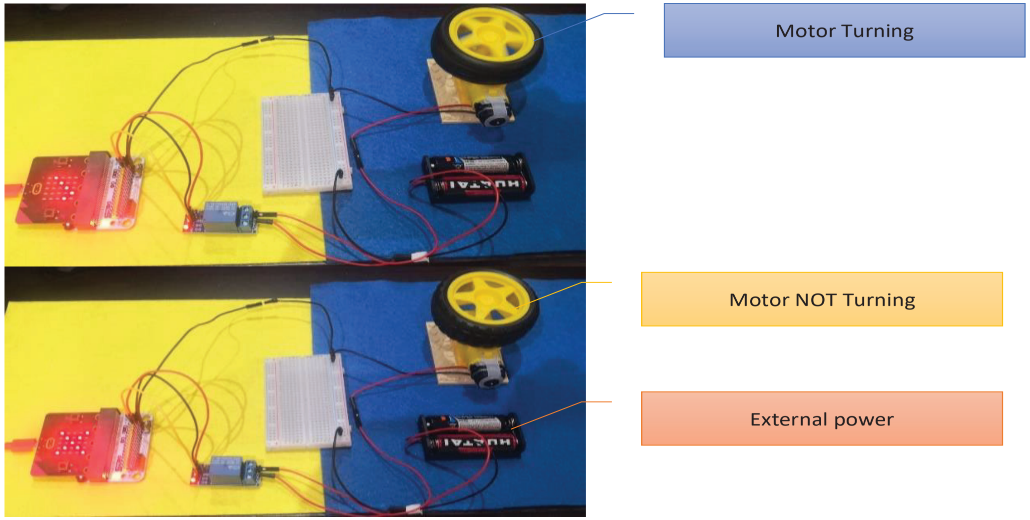
Content (Grade 9 / Term 1)

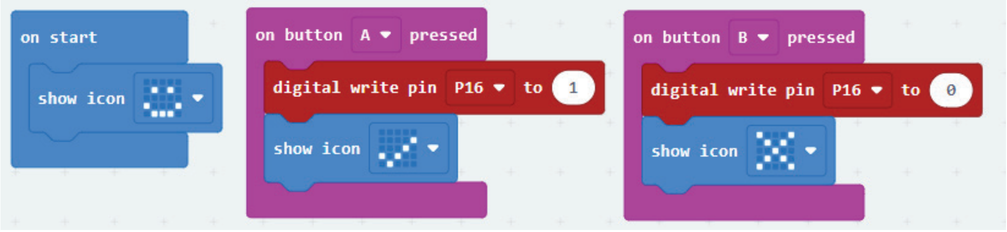
Example activity with a relay

In the example below a DC motor is connected to an external power source, a relay on the power source. A signal is sent using pin 16 on the microcontroller to control NC pin. M



is used to switch the relay via the



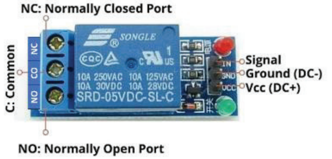


```
on start
  show icon

on button A ▾ pressed
  digital write pin P16 ▾ to 1
  show icon

on button B ▾ pressed
  digital write pin P16 ▾ to 0
  show icon
```

Notes/Examples

IMPORTANT in GRADE 9 all robotics / physical computing projects is constructed to interact with onboard sensors and onboard output components of the microcontroller. A breakout board/expansion board and a maximum of FOUR additional hardware components (Actuators e.g., Servo's, DC Motors AND Sensors etc. can be connected to the Microcontroller via a breakout board). In addition, simple electronic components such as LED's, Potentiometers, LDR's, DC motors or (DC water pumps) and Buzzers may be added to enhance the project.
A motor driver board can also be introduced conceptually (NOT for assessment purposes)
All connections with crocodile clips (or derivatives) and jumper cables only (NO soldering)

NOTE:
The learners need to understand the usage op the pins



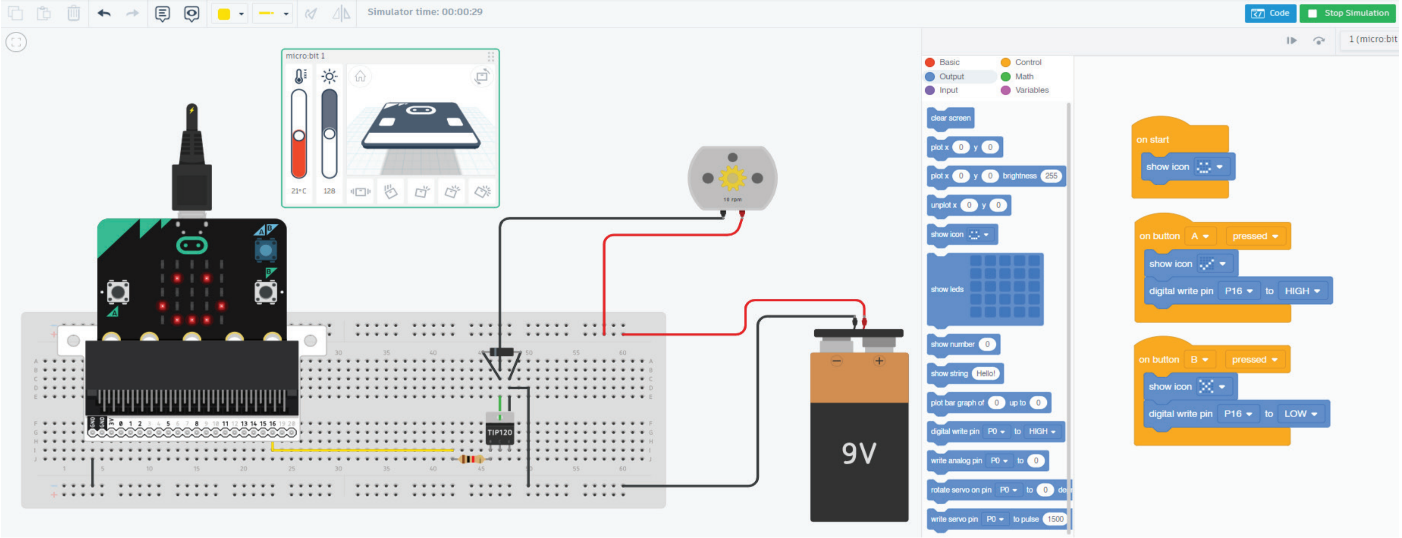
Six essential points about the SRD-5VDC-SL-C relay's pins:

- Coil+ (VCC): Positive terminal for energizing the relay coil (5V DC).
- Coil- (GND): Negative terminal for completing the coil circuit.
- Signal/Input (SIG or IN): Optional input for external control, activating the relay based on applied signals.
- Common (COM): Common terminal connecting to either NO or NC contacts.
- Normally Open (NO): One switch terminal, open when the relay is energized.
- Normally Closed (NC): The other switch terminal, closed when the relay is energized.

Content (Grade 9 / Term 1)

Example with a transistor

In the example below a DC motor is connected to a 9 V battery and a TIP120 Darlington NPN transistor. A general purpose 1N4001 diode and a 1 Kilo Ω resistor. The diode is there to prohibit any damage and feedback from the DC motor. When pin 16 is passed a current, the transistor allows for current to flow through the collector emitter junction. The emitter is connected to GND and the motor to the collector.

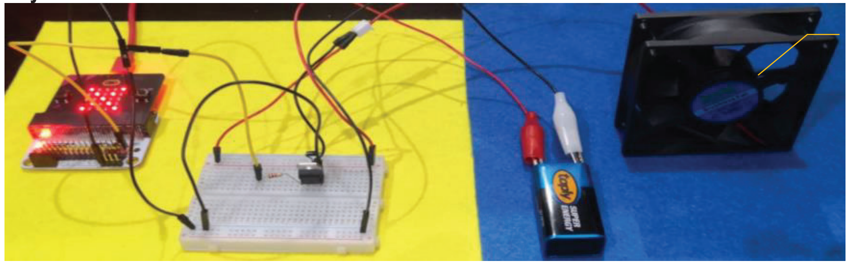


on start
show icon

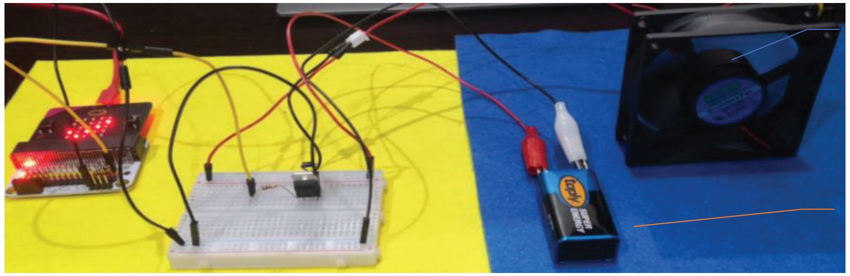
on button A pressed
show icon
digital write pin P16 to HIGH

on button B pressed
show icon
digital write pin P16 to LOW

Physical build



Motor NOT Turning

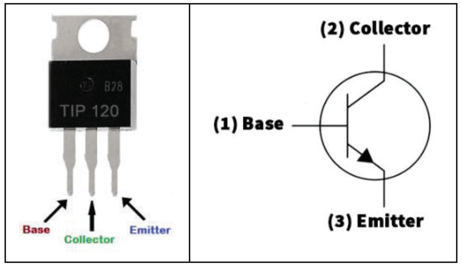


Motor Turning

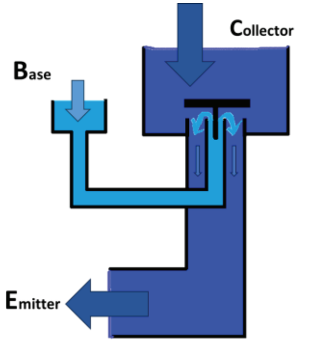
External power

Notes/Examples

The learners need to understand that transistors act as switches.

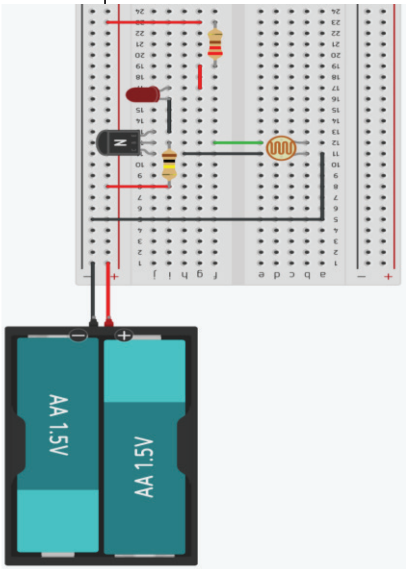
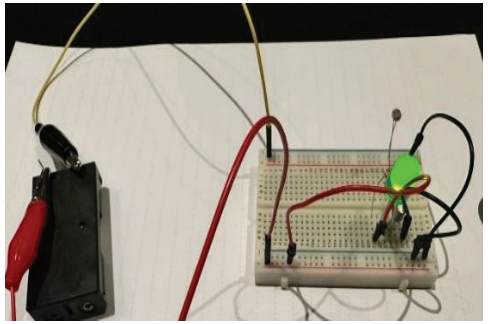
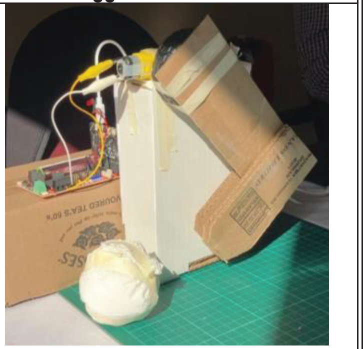


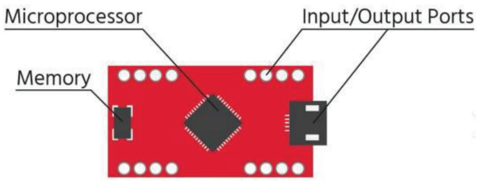
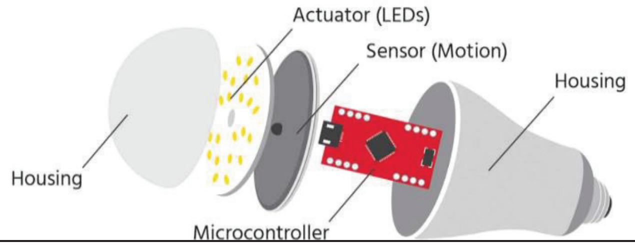
(In the picture below a small little current of water opens a valve to allow a large stream of water through.)



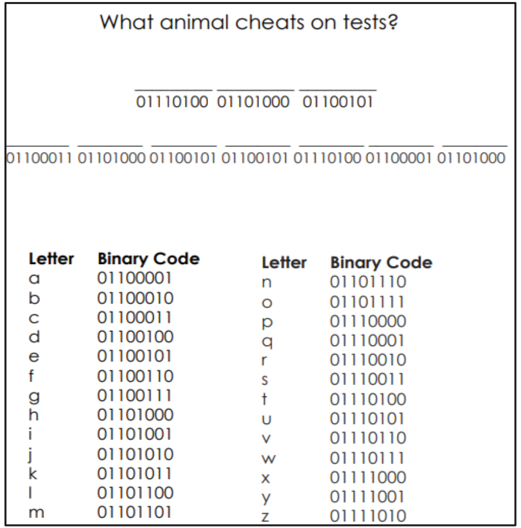
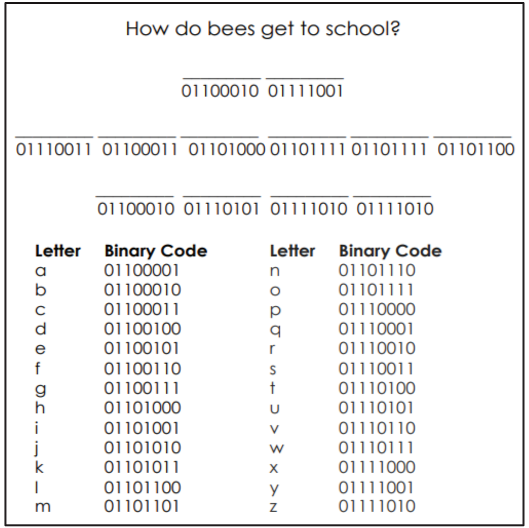
In a transistor the base acts as input switch, only a small amount of current is required to which on the transistor to allow current to flow through the collector emitter junction.

IMPORTANT Schools that do have computer laboratories may deploy the use of simulation environments (or software for reinforcement)

Content (Grade 9 / Term 1)	Notes/Examples
Extra simple transistor circuit In the example below an NPN transistor is used in a circuit to switch on an LED based on the light intensity from an LDR  Sample build 	
R.5 Design a simple artefact based on a set of design instructions	
R.6 Mimic the operations of a robot	**Link to R.5 and Cs**
R.7 Create, test and execute a set of robotic instructions	**R.5, R.6 and R.7 done together**
Example activity – project **Experiment / Project**  The learners must be able to apply the principles of design thinking to develop an artefact where a relay or transistor is used as a switch to allow a component such as a fan, DC water pump etc., to operate based on input from a signal. One such example is an automatic water dispenser, triggered by an ultrasonic sensor, or a sound triggered ball kicker. **Automatic water dispensor**  **Sound triggered ball kicker** 	**Link to R.6 and R.7** IMPORTANT **IMPORTANT** Inform the learners about the final term project. The learners should be able to work and plan their project, throughout the duration of the year.

Content (Grade 9 / Term 1)	**Notes/Examples**
Digital Concepts	
D.1 Outline the concept of technology and purpose of information technology (IT)	**Link to D.3, and D.7 and C.2 and R.5**
Reinforce from the previous grades using different examples and activities. Learners should be able to: • explain which devices are used in the context of information technology • relate the concept of computing devices to that of an IT tool, e.g. computer, microcontroller **Technology and Networks:** ○ **Technology**: Technology refers to the application of scientific knowledge, tools, and techniques to create solutions, products, and systems that enhance human capabilities and improve the quality of life. ○ **Networks**: Networks are a key component of technology, representing interconnected systems that facilitate the transmission of data and communication between devices or users. ○ **Purpose of Information Technology (IT) and Networks:** ○ **Information Technology (IT)**: IT encompasses the use of computers, software, networks, and other digital technologies to manage and process information efficiently. **Why IT and Networks Matter:** • They help people and organisations communicate, share data, and work together. • They store and protect information. • They enable online shopping, banking, and more. • They come with challenges like security and privacy. **Future Trends:** IT keeps evolving with enhancements like cloud computing and faster internet, making our lives and work even more connected.	Learners need to understand • the role of technology and information technology relating to networks • purpose of networks
D.2 Recognise that he or she is living as citizens in a digital world.	**Done with D.4.**
Reinforce from the previous grades using different examples and activities. Address the challenges related to network reliability, cybersecurity, and ethical considerations, including privacy and safety issues **Example activity – Cyber security** Learners work in pairs. Each pair watch the following videos https://youtu.be/vvsfM5Dixow https://youtu.be/inWWhr5tnEA https://youtu.be/Q2W_nNdReXM and complete a KWLS chart. Learners report back on what they have learned.	Learners need to understand • what cybersecurity is • how to prevent threats / protect themselves
D.3 Demonstrate an understanding of the concept of a computing device.	**Link to D.1, D.4 and D.7, C.1 – C.5 and R.5 – R.7**
Reinforce from the previous grades using different examples and activities. A microcontroller is often used in Internet of Things (IoT) applications because it is inexpensive, has low power consumption, and can easily integrate into various devices. A microcontroller is a single chip that contains a processor, memory, and input/output parts. The processor runs instructions, the memory stores data and programs, and the I/O parts let the microcontroller communicate with the environment using sensors and tools. A microcontroller is therefore a small computer typically also used in embedded systems, such as home appliances to control electronic devices. An IoT device would most likely use a microcontroller as its brain **Simple microcontroller**  **Anatomy of a *Thing* (a globe with microcontroller)** 	Learners need to understand that a microcontroller also serves as a computing device as part of the Internet of Things (IoT) Internet of Things (IoT) devices, such as smart home systems, wearables, and environmental sensors, use microcontrollers to connect to the internet and perform various functions. An IoT device would most likely use a microcontroller as its brain

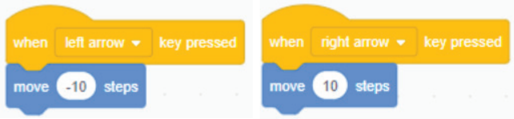
Content (Grade 9 / Term 1)	Notes/Examples
D.4 Identify the common uses of ICT in the real world	**Done with D.1, D.2.**
D.5 Differentiate between the components of an ICT system	**Link to D.4, D,7 and C.7, R.3 – R.7**
Reinforce from the previous grades using different examples and activities. Revision of basic network components. Networks consist of various connected (wired or wireless connections) components that work together to enable data communication and resource sharing. Computing devices in a network include: • Computers (suing microprocessors) /IoT devices (using e.g. microcontrollers) • Server • Router • Modem • Gateway	**NOTE:** No need to explain protocols, learners only need to understand at a high level that a gateway allows communication. Explain the simple concept of a gateway e.g. a bridge to connect two countries.
D.6 Explain how the adaptation of technology impacted the world we work and live in	**Link to D.2**
Revise important impact of networks and Internet. • Global connectivity • Communication and Collaboration • Information Access • Education and E-Learning • Entertainment and Media • Smart Technologies • Cybersecurity Challenges • Privacy Concerns Revise how networks provide access to a vast amount of information and resources available on the internet, benefiting education, research, and entertainment.	
D.8 Interpret a pattern to represent or communicate a message or image	**Link to D.9 and C.1 – C.5 and R.5 – R.7**
Reinforce from the previous grades using different examples and activities Revision of binary logic. Create a quiz using online apps, such as Kahoot, to revise binary logic. This can also be done as a worksheet. **Example activity:** Create a few binary logic puzzles on a separate sheet of paper. These puzzles could involve completing patterns or sequences with binary digits., e.g.	

Content (Grade 9 / Term 1)	Notes/Examples
 What animal cheats on tests? 01110100 01101000 01100101 01100011 01101000 01100101 01100101 01110100 01100001 01101000 Letter Binary Code Letter Binary Code a 01100001 n 01101110 b 01100010 o 01101111 c 01100011 p 01110000 d 01100100 q 01110001 e 01100101 r 01110010 f 01100110 s 01110011 g 01100111 t 01110100 h 01101000 u 01110101 i 01101001 v 01110110 j 01101010 w 01110111 k 01101011 x 01111000 l 01101100 y 01111001 m 01101101 z 01111010  How do bees get to school? 01100010 01111001 01110011 01100011 01101000 01101111 01101111 01101100 01100010 01110101 01111010 01111010 Letter Binary Code Letter Binary Code a 01100001 n 01101110 b 01100010 o 01101111 c 01100011 p 01110000 d 01100100 q 01110001 e 01100101 r 01110010 f 01100110 s 01110011 g 01100111 t 01110100 h 01101000 u 01110101 i 01101001 v 01110110 j 01101010 w 01110111 k 01101011 x 01111000 l 01101100 y 01111001 m 01101101 z 01111010	
D.9 Create a pattern to represent or communicate a message or image	**Link to D.8.**
Continue with the previous activity and allow learners to create their own quizzes and share it.	
D.10 Demonstrate a basic proficiency in the application of digital skills.	**Link to C.2 – C.5 and R.5 – R.7**
Example activity: Begin by discussing the importance of networks and file sharing in today's digital world. Explain that networks enable us to access files and resources on other computers or devices connected to the same network. Provide a few real-life examples of when we might need to access files or print to a networked printer, such as in a school, office, or home setting. Review the Networking Basics: Use the whiteboard to illustrate these concepts with simple diagrams and explanations. Group activity: 1. You oversee a school project, and you need to access research materials stored on a teacher's computer in the school's network. 2. You have an assignment that needs to be printed, but your home printer is not working. You want to send the document to your neighbour’s networked printer for printing. Each group to brainstorm and design an activity plan that addresses their scenario considering: - What tools or equipment do they need? - How will they connect to the network (Wi-Fi or Ethernet)? - What steps will they follow to access files or print to the printer? - Are there any security considerations to keep in mind? (Be creative in the solutions while considering the practicality of their plans.) Presentation: Group presents activity plan to the class with Q&A (provide feedback and suggest improvement.) Class Discussion: Importance of proper planning when it comes to networked activities. Emphasise respecting privacy and security when accessing files or using shared resources on a network.	

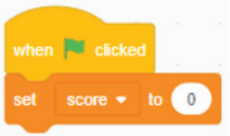
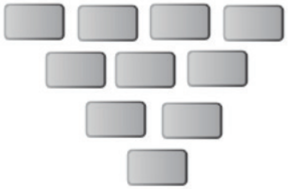
3.3.2 Term 2

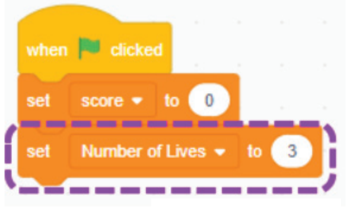
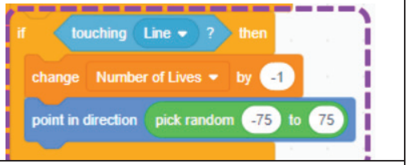
<table>
<tr><th>Content (Grade 9 / Term 2)</th><th>Notes/Examples</th></tr>
<tr><th>C.1 Apply computational thinking (CT) skills to develop a set of logical instructions to solve a problem.</th><th>Link to C.2 – C.7 and R.5 – R.7</th></tr>
<tr><td>

Example activity – Breakout game
A breakout game consists of one paddle used to return a bouncing ball back and forth across the screen. The aim of the game is to break the bricks of a brick wall by getting the ball to hit/bounce on the bricks. breakout game in scratch | 101 computing
In pairs, work through the code below and create your own breakout game like the one below:

<table>
<tr><th>Step1: Paddle</th><th>Step 2:</th></tr>
<tr><td>Use paddle sprite 
Add code to control paddle

when left arrow ▾ key pressed
move -10 steps
when right arrow ▾ key pressed
move 10 steps</td><td>Add ball sprite Resize to 50 
Add code for ball sprite

when 🏴 clicked
point in direction (pick random -75 to 75)
forever
move 10 steps
if on edge, bounce</td></tr>
<tr><th>Step 3: Bouncing off paddle</th><th>Step 4: Make variable for keeping score</th></tr>
<tr><td>Add code to ball sprite to make it bounce when it hits the paddle.

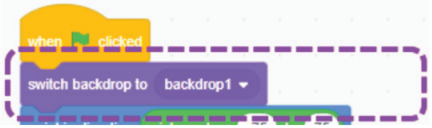
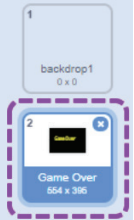
if touching Paddle ▾ ? then
point in direction (pick random -75 to 75)</td><td>Make variable 
Add code for paddle sprite to initialize score to 0 before game starts

when 🏴 clicked
set score ▾ to 0</td></tr>
<tr><th>Step 5: Add brick sprite</th><th>Step 6:</th></tr>
<tr><td>Use button 3 for brick. 

Duplicate brick to form 10 bricks.

when 🏴 clicked
show
forever
if touching Ball ▾ ? then
hide
change score ▾ by 1
Add code for brick to disappear when the ball hits it and increment score by 1 when it happens.</td><td>Create a variable, Number_of_Lives for paddle sprite and initialize it to 3 when the game starts.

when 🏴 clicked
set score ▾ to 0
set Number of Lives ▾ to 3
Add a line sprite and position it below the paddle. 
Add code to the ball sprite to see if it hits the lifeline (if it does, the player will lose a life and the ball will bounce back.

if touching Line ▾ ? then
change Number of Lives ▾ by -1
point in direction (pick random -75 to 75)</td></tr>
<tr><th>Step 7: Add Game Over backdrop</th><th></th></tr>
</table>

</td><td>

Note:
Learners use computational, what they have learned in coding as well as their experience from working through the example game to create their own game.
This activity includes C..2, (creating code) C.3 (interpreting code), C.4 (debugging code) and C.5 (evaluating their code for improvement)

</td></tr>
</table>

Content (Grade 9 / Term 2)	Notes/Examples
Add Game Over backdrop Switch to Game Over backdrop  when [flag] clicked switch backdrop to backdrop1  Load the program Execute it and ensure that you understand the code.	
C.2 Present a simple coding solution using symbolic or written statements representing sequences of commands, single repetition, and conditional constructs.	**Link to C.1, C.3-C.7 and R.5 – R.7**
Activity 1: Scratch Scrabble score keeper level 1 (String handling, determine length of a string, extract characters in a string) In the game of Scrabble, you get a score according to the letters in the word that you place on the board. The purpose of this program is to calculate the total for each word and keep the score by adding the total of all the words. Input a word and calculates the score for a word. The letters are scored as follows: A = 1; B=2; C=3; D=4; etc. If a word is longer than 6 letters, the first and last letters must count double points. Examples: • The word "computer" will score 132 (c=3x2)+(o=15)+(m=13)+(p=16)+(u=21)+(t=20)+(e=5)+(r=18x2) • The word "strong" will score 93 (s=19)+(t=20)+(r=18)+(o=15)+(n=14)+(g=7) Add the points that the player got to the player's total. For each word, display the points for the word as well as the player's total at that stage The program must allow the user to repeatedly place a word until the user has at least 100 points. The program must also count how many words it took to reach a total of at least 100 points and display the number of words. **Activity 2: Scratch Scrabble score keeper level 2:** Expand the level 1 Scratch Scrabble score keeper for two Sprites to play against each other. • The program inputs the name for each player. • Each player gets a turn to choose a word (input a word). • The first player who has a score of 100 points, wins the game. (The game will continue until a player has a total of at least 100 points.) • If a player scores more than 50 for a word, the other player must say "Wow, well done!" • Each Sprite's total points must be updated each time. • The Sprite who has at least 100 points first, wins the game. Display the total scores for both players as well as the name of the winner.	
C.3 Interpret and execute a given symbolic or written set of commands	**Link to C.1 and C.2**
C.4 Debug a given symbolic or written set of instructions	**C.3 and C.4 are done with C.1 and C.2**
Done with C.1 and C.2	
C.5 Evaluate a given solution towards potential improvement	
Example activity – Evaluate two programs with the same outcome Two sprites represent a program to break a number up into, e.g. thousands, hundreds, tens and ones, e.g. 6789 is displayed as 6000 + 700 + 80 + 9. However, the two programs use different approaches. In pairs, study the programs, describe and explain the method followed by each. Use a tractable to work through each program. • Will this program work for any number (even bigger than 9999)? If yes, what needs to be changed to make it work for any number? • Evaluate each of the programs in terms of effectiveness. • Pairs report back and share their discussions and findings.	**Note:** Learners may need guidance on this activity. Let pairs use a traceable for each program to describe and explain how each program work. There are similarities and differences between the programs. Pairs should answer the following questions after working through the two programs:

Content (Grade 9 / Term 2)

Program 1

```
when this sprite clicked
delete all of [iNumberValue ▾]
set [iNumber ▾] to (pick random (10) to (9999))
say (join [Number is] (iNumber)) for (2) seconds
set [iPlaceValue ▾] to (1)
repeat until <(iNumber) = (0)>
    set [iDigit ▾] to ((iNumber) mod (10))
    set [iDigitValue ▾] to ((iDigit) * (iPlaceValue))
    set [iDigit ▾] to ((iNumber) mod (iPlaceValue))
    add (iDigitValue) to [iNumberValue ▾]
    set [iPlaceValue ▾] to ((iPlaceValue) * (10))
    set [iNumber ▾] to ([floor ▾] of ((iNumber) / (10)))
end
say (join (item (4) of [iNumberValue ▾]) (+)) for (2) seconds
say (join (item (3) of [iNumberValue ▾]) (+)) for (2) seconds
say (join (item (2) of [iNumberValue ▾]) (+)) for (2) seconds
say (item (1) of [iNumberValue ▾]) for (2) seconds
```

Program 2

```
when this sprite clicked
delete all of [iNumberValue ▾]
set [iNumber ▾] to (pick random (10) to (9999))
say (join [Number is:] (iNumber)) for (2) seconds
set [iPlaceValue ▾] to (1)
set [iDigit ▾] to (0)
set [iPosition ▾] to (length of (iNumber))
repeat (iPosition)
    set [iDigit ▾] to (letter (iPosition) of (iNumber))
    set [iDigitValue ▾] to ((iDigit) * (iPlaceValue))
    add (iDigitValue) to [iNumberValue ▾]
    set [iPlaceValue ▾] to ((iPlaceValue) * (10))
    change [iPosition ▾] by (-1)
end
say (join (item (4) of [iNumberValue ▾]) (+)) for (2) seconds
say (join (item (3) of [iNumberValue ▾]) (+)) for (2) seconds
say (join (item (2) of [iNumberValue ▾]) (+)) for (2) seconds
say (item (1) of [iNumberValue ▾]) for (2) seconds
```

Notes/Examples

- When evaluating the programs, answer the following questions
- How is the input to both programs different/the same?
- How is the output of both programs different/the same?
- How are the loops (repeat structures) different/the same?
- For the same number, how many times will the loop in each program execute?
- Which of the two programs will work with any number (even bigger than 9999)?

Note:
It is important that learners understand that, in many instances, a problem can be solved using different approaches and/or tools.

C.6 Recognise and interpret patterns in symbolic sets of data or visualisations.

Link to C.2 – C.5 and C.7, D.8 and D.9

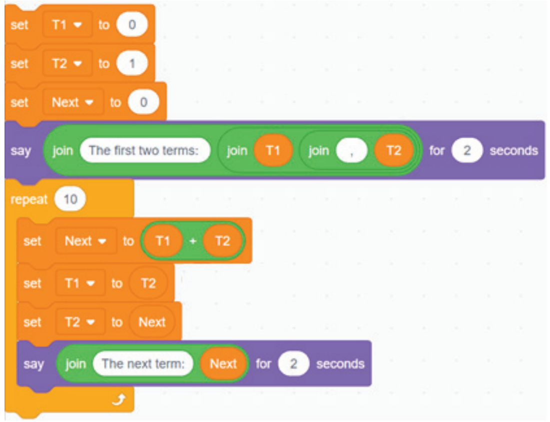
C.7 Create or complete a pattern to represent a data set

Do with C.6

Example activity – Fibonacci numbers

Fibonacci numbers start with two predetermined terms and each term afterwards is the sum of the preceding two terms.
The first two numbers are 0 and 1 ($T_1 = 0$, $T_2 = 1$) and the next number is always the sum of the previous two numbers
The first few Fibonacci numbers with signature (0,1) are 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144…
(**Note**: The first two terms are the signature. The signature may vary.)

- Work through the program on the right to see if it displays the first 10 Fibonacci numbers correctly.
- Now, change the signature (the first to numbers to e.g., 1 and 2 and run the program to see what happens.

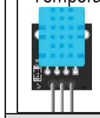


```
set [T1 ▾] to (0)
set [T2 ▾] to (1)
set [Next ▾] to (0)
say (join [The first two terms:] (join (T1) (join [,] (T2)))) for (2) seconds
repeat (10)
    set [Next ▾] to ((T1) + (T2))
    set [T1 ▾] to (T2)
    set [T2 ▾] to (Next)
    say (join [The next term:] (Next)) for (2) seconds
end
```

C.6 and C.7 done together

Content (Grade 9 / Term 2)	Notes/Examples
Robotics	
R.3 Outline the different components of a robot	**Link to R.5 – R.7**

Learners need to describe and use the following sensors:

Component	Basic definition	Purpose	Practical application example – May be used to:
Soil Moisture Sensor 	A sensor that measures the moisture content in so	Employed in agricultural or gardening robots to monitor soil conditions for watering or plant care.	Monitors soil conditions for automated plant care systems in agricultural robots.
Temperature and Humidity Sensor: 	A sensor that measures both temperature and humidity levels in the environment.	Enables climate control, environmental monitoring, and weather reporting in robots.	Enables climate control, environmental monitoring, and weather reporting in robots.

Introduce learners to sensors that provide analogue input, e.g. soil moisture sensor, temperature sensor and humidity sensor

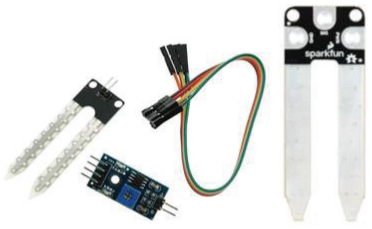
Learners need to know the elementary difference between digital and analogue input and how to process the readings in block-based code

R.5 Design a simple artefact based on a set of design instructions

R.6 Mimic the operations of a robot	**Link to R.5 and C.1 – C.7**

Example activity - Soil moisture artefact with analogue sensor
For version 2 (as opposed to the Grade 7 version), the learners are introduced to the concept of Analogue input (values 0 to 1023) vs Digital input (Binary – On and Off / 0 or 1)
For this version learners substitute the simple probes with a soil moisture sensor.

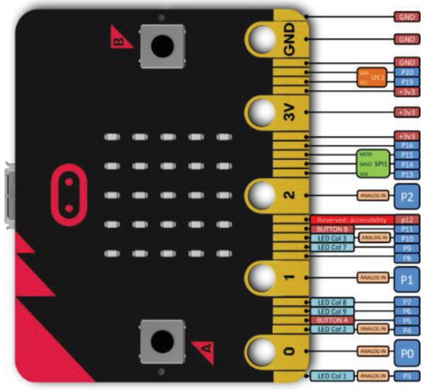
Soil moisture with sensor and analogue input
https://www.tinkercad.com/blog/tinkercads-new-soil-moisture-sensor



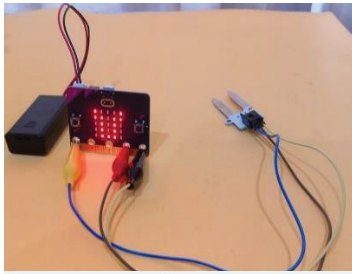
NOTE:
Analogue input pins on a microcontroller allow it to measure continuous voltage levels, making it suitable for tasks like reading sensor data such as light or temperature. Digital input pins, on the other hand, are used for detecting binary signals like button presses or switch states, making them ideal for user interactions and simple on/off sensing.

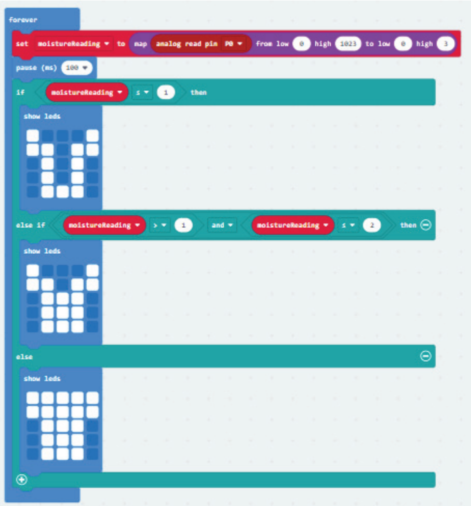
Analogue input pins on a microcontroller typically have a range of 0 to 1023, representing the full spectrum of voltage levels they can measure, which makes them versatile for a wide range of sensor applications.

Also see: Pins on the Micro: bit

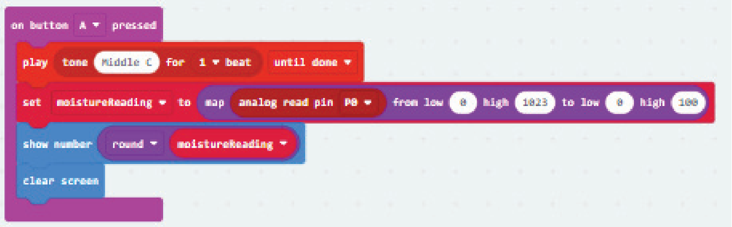


Content (Grade 9 / Term 2)	Notes/Examples





```
forever
  set moistureReading ▾ to map analog read pin P0 ▾ from low 0 high 1023 to low 0 high 3
  pause (ms) 100 ▾
  if moistureReading ▾ ≤ ▾ 1 then
    show leds
  else if moistureReading ▾ > ▾ 1 and ▾ moistureReading ▾ ≤ ▾ 2 then ⊖
    show leds
  else ⊖
    show leds
  ⊕
```

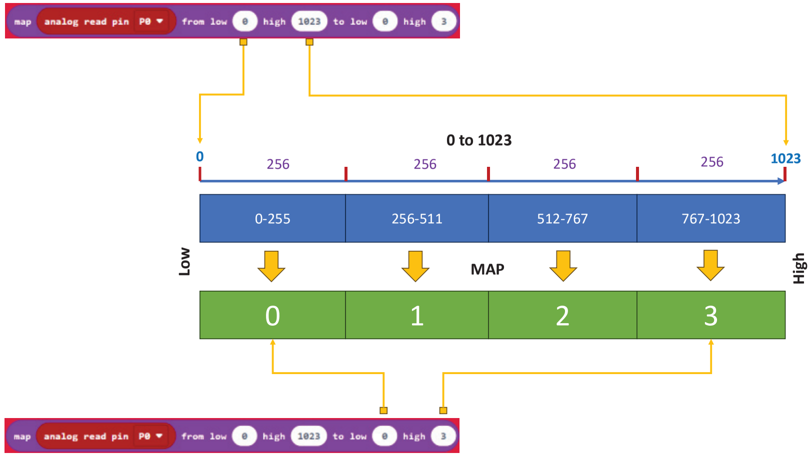


```
on button A ▾ pressed
  play tone Middle C for 1 ▾ beat until done ▾
  set moistureReading ▾ to map analog read pin P0 ▾ from low 0 high 1023 to low 0 high 100
  show number round ▾ moistureReading ▾
  clear screen
```

Also see: (https://makecode.microbit.org/55671-93208-82700-45427) and https://makecode.microbit.org/reference/pins/map

In relation to the code the learners also need to know how to use the MAP function as part of the block-based language used to map ranges to single values from readings.

The Map function in MakeCode (mapping analogue data to digital data)



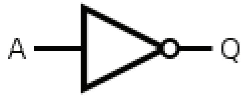
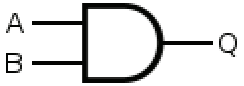
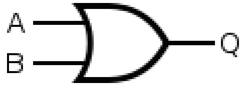
Content (Grade 9 / Term 2)	Notes/Examples

Revisit logical gates and truth tables

The learners should present a basic understanding of the principles of:

- Boolean logic and
- Logical gates (AND, OR and NOT) and how it relates to the design, programming, and functioning of robotic systems.

The learners revisit the 3 ANSI symbols for the three logical gates.

NOT	AND	OR
		

The learners should be able to:

- represent a simple truth table for each of the gates.
- interpret a simple logical gate diagram comprising of no more than 3 inputs and a maximum of four gates to determine a single outcome. (Either true or false)
- complete a partially completed truth table.
- present a basic discussion of the operation of the circuit as well and its practical application, with real-world scenarios.
- setup and draw a simple logical gate circuit to present a real-life problem involving sensors (simple circuit with a maximum of three inputs) and a maximum of four gates.

Also see:

https://en.m.wikipedia.org/wiki/Logic_gate
https://learnlearn.uk/gcsecs/logic-gates/
https://www.bbc.co.uk/bitesize/guides/zkkkw6f/revision/1
http://www.bitsofbytes.co/exam-questions---logic-gates.html

Examples of scenarios

Scenario 1

Smart home application that automatically turns on a light bulb (actuator) in a room when it gets dark, but only if there is someone in the room. You have two sensors to accomplish this: a light sensor and a motion sensor.

Example

An AND gate outputs a high (1) signal only when both of its inputs are high. In this context, we can use an AND gate to combine signals from both sensors, ensuring that the light turns on only when it's dark (low light sensor reading) and there's motion (motion sensor detects someone).

- Light Sensor (LS) -> Input 1
- Motion Sensor (MS) -> Input 2
- AND Gate Output (Y) -> Actuator (Light Bulb)

LS	MS	Y (Light Bulb)
0	0	0 (Off)
0	1	0 (Off)
1	0	0 (Off)
1	1	1 (On)

Scenario 2

A parking lot entry gate with a boom gate and a ticket payment system. The gate should only open if two conditions are met: a customer has inserted a valid parking ticket, and a sensor detects a vehicle waiting to exit.

The ticket payment system (TPS) verifies if a customer has inserted a valid parking ticket. It provides a binary signal (0 for invalid ticket, 1 for valid ticket).
The vehicle sensor (VS) detects the presence of a vehicle waiting to exit and provides a binary signal (0 for no vehicle, 1 for a vehicle detected).
The AND gate combines these signals. It outputs 1 only when both conditions are met (valid ticket and vehicle detected).
The NOT gate inverts the AND gate's output. If the AND gate outputs 1, the NOT gate will output 0 (opening the boom gate); if the AND gate outputs 0, the NOT gate will output 1 (keeping the boom gate closed).

Content (Grade 9 / Term 2)

Notes/Examples

TPS	VS	Y (Boom Gate)
0	0	0 (Off)
0	1	0 (Off)
1	0	0 (Off)
1	1	1 (On)

A NOT gate inverts its input signal. You can use a NOT gate to ensure the boom gate remains closed when the AND gate's output is 0 (i.e., both conditions are not met).

AND Gate Output (Y) -> Input
NOT Gate Output -> Boom Gate Control

Y (Boom Gate)	NOT(Y) (Boom Gate)
0	1 (Open)
1	0 (Closed)

Circuit Operation:

The ticket payment system verifies if a customer has inserted a valid parking ticket. It provides a binary signal (0 for invalid ticket, 1 for valid ticket).
The vehicle sensor detects the presence of a vehicle waiting to exit and provides a binary signal (0 for no vehicle, 1 for a vehicle detected).
The AND gate combines these signals. It outputs 1 only when both conditions are met (valid ticket and vehicle detected).
The NOT gate inverts the AND gate's output. If the AND gate outputs 1, the NOT gate will output 0 (opening the boom gate); if the AND gate outputs 0, the NOT gate will output 1 (keeping the boom gate closed).

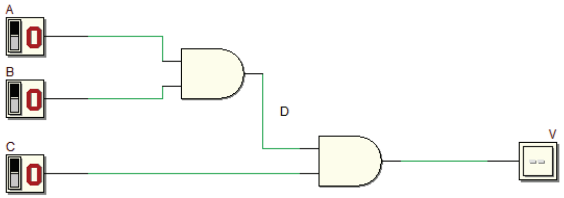
Application:

- A customer inserts a valid parking ticket (TPS=1).
- A vehicle is detected by the sensor (VS=1).
- The AND gate receives both inputs as 1 and outputs 1.
- The NOT gate inverts the AND gate's output, resulting in a 0.
- The boom gate is controlled by the NOT gate's output, so it opens.
- This example illustrates how logical gates and circuits can be used to control access to a parking lot by ensuring that customers pay for a ticket before the boom gate opens, enhancing security and revenue collection.

Scenario 3

Imagine you have a security system for a vault that requires the correct combination of three inputs (A, B, and C) to unlock the vault door. You want to use logical gates to control whether the vault door V is open or closed.

A	B	C	D (A AND B)	V (Vault) (C AND C)
0	0	0	0	0 (Closed)
0	0	1	0	0 (Closed)
0	1	0	0	0 (Closed)
0	1	1	0	0 (Closed)
1	0	0	0	0 (Closed)
1	0	1	0	0 (Closed)
1	1	0	1	0 (Closed)
1	1	1	1	1 (Open)

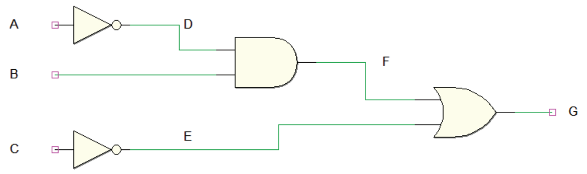


Example of a truth table and a logic circuit (with 3 inputs and a maximum of 4 gates)

Content (Grade 9 / Term 2)

Notes/Examples

A	B	C	D (NOT A)	E (NOT C)	F (D OR B)	G (F AND E)
0	0	0	1	1	1	1
0	0	1	1	0	1	0
0	1	0	1	1	1	1
0	1	1	1	0	1	0
1	0	0	0	1	0	0
1	0	1	0	0	0	0
1	1	0	0	1	1	1
1	1	1	0	0	1	0



Experiments / Projects

The learners must be able to apply the principles of design thinking to develop an artefact where a relay or transistor is used as a switch and a sensor to activate a device such as a DC water pump. Such an experiment or project could be a self-watering plant or an automatic biltong drier with humidity sensor.

Source (https://www.instructables.com/Automatic-Plant-Watering-System-Using-a-Microbit/)

Automatic Biltong Drier, where a humidity sensor and temperature sensor regulates the fan.

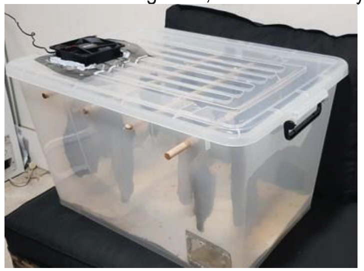


Image source: https://preview.redd.it/8uxo4lob0tk41.jpg?width=1080&crop=smart&auto=webp&s=a90c517242dfb64eb14a3756f18cf0a5c7cb9d9e

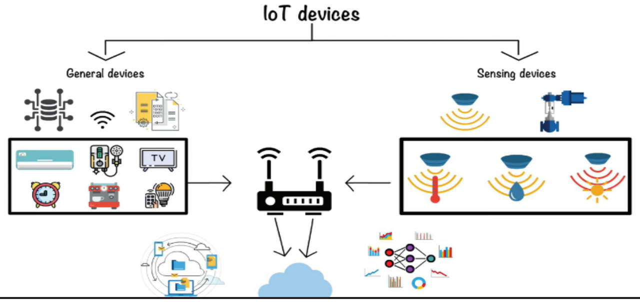
R.7 Create, test and execute a set of robotic instructions

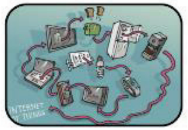
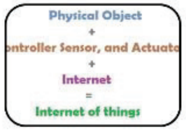
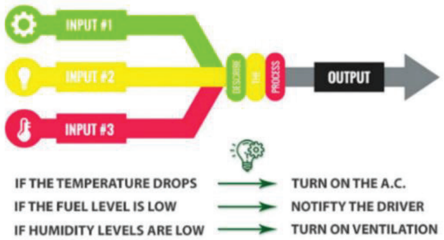
Example activity 1 – Flappy bird game

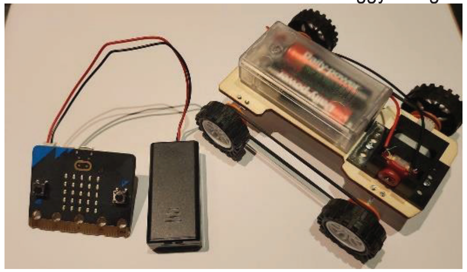
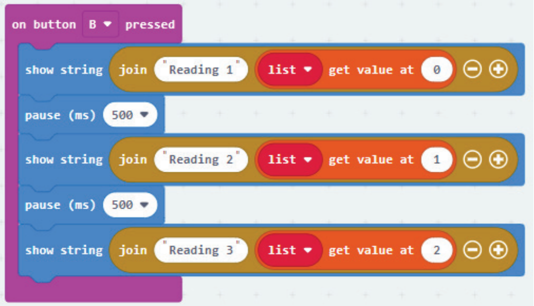
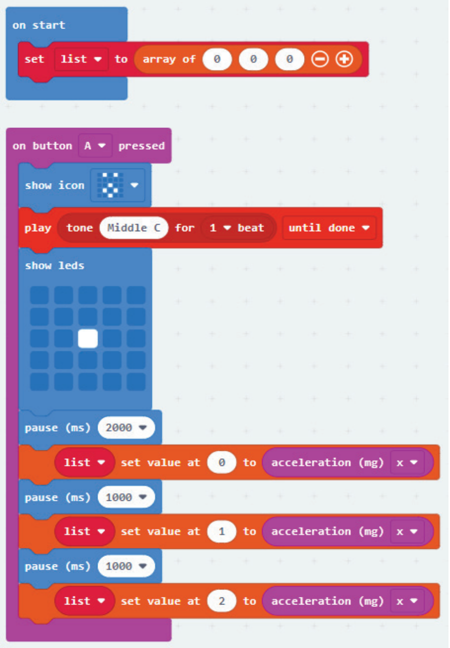
The learners need to create a game that is based on the Flappy Bird game. The “object” (bird) must fly through open spaces in a maze.

How the game must work:

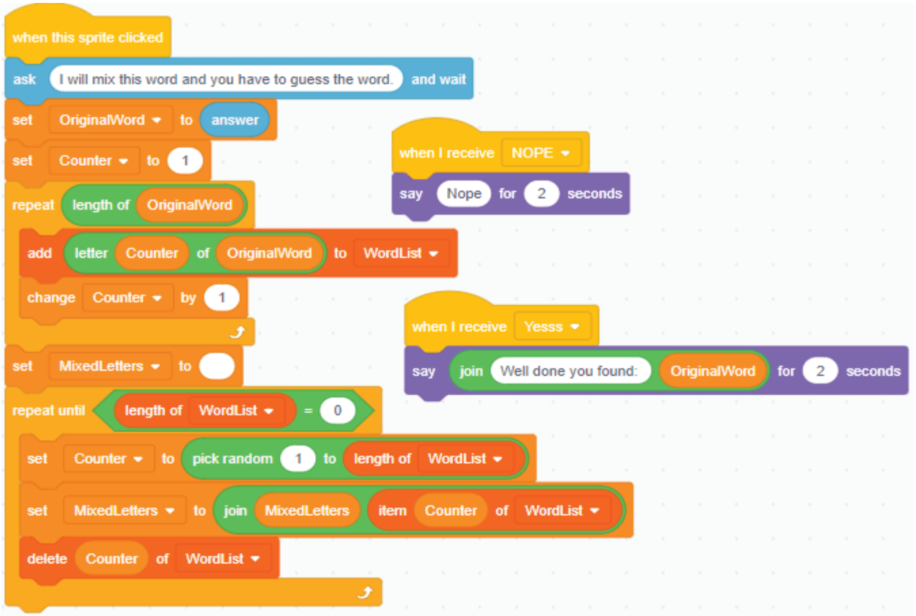
- When both buttons A and B are pressed, a maze must be displayed, with the “bird” in the bottom left corner.
- One LED will act as the “bird” and will start in the bottom left corner. The “bird” LED must display with a higher intensity than the other LED lights.
- Generate a random pattern of LED’s displayed on the screen to simulate a maze. These LED’s must display in a lower intensity than the “bird”.
- When button B is pressed the “bird” must navigate through the maze until it reaches the Y-coordinate of 4 (the fifth column).
- When the “bird” hits a wall (or an LED light), an “X” must be displayed to indicate “game over”.

Content (Grade 9 / Term 2)	**Notes/Examples**
Digital Concepts	
D.1 Outline the concept of technology and purpose of information technology (IT)	**Done with D.2 and D.3.**
Introduce the Internet of Things (IoT) The Internet of Things (IoT) refers to a network of interconnected physical devices or "things" that are embedded with sensors, software, and other technologies to collect and exchange data over the internet or other communication networks. These devices can range from everyday objects such as online cameras, security devices, vehicles, and wearable gadgets to industrial machines and infrastructure components. IoT relies on infrastructure, including networks, servers, and cloud computing, to enable data communication and processing. IT systems provide the backbone for IoT to function effectively. **Example activity – Introduction to IoT** Divide learners into small groups/pairs Learners watch the following videos https://youtu.be/ps9ucSDH8s4 and https://youtu.be/6mBO2vqLv38 Each leaner completes a KWLS chart and report back	Introduction to Internet of Things (IoT) Link to the concept of networks Learners need to understand • What the IoT is • Basic components of IoT • Computing devices that enable the IoT • Challenges with IoT such as privacy and security
D.2 Recognise that he or she is living as citizens in a digital world.	**Done with D.4, D.6.**
Ethical implications of IoT include security, privacy, and ethical considerations. Example activity – IoT, cybersecurity and privacy In pairs learners watch the following video https://youtu.be/4XycTona5ql and https://youtu.be/u1ymmRQ_p3k and complete a KWLS chart Discuss issues related to IoT: • Cybersecurity: IoT devices are vulnerable to cyberattacks • Privacy and data collection: IoT devices collect vast amounts of data, often including personal information. Users may not always have control over the data generated by IoT devices. Discussion of who collects and owns this data.	Link to D.1
D.3 Demonstrate an understanding of the concept of a computing device.	**Done with D.1.**
D.4 Identify the common uses of ICT in the real world.	
Describe the concept of the Internet of Things (IoT) and explain how it connects physical devices and data. Common uses of IoT: • Smart Homes: IoT helps control lights, thermostats, and security in homes. • Healthcare: It's used for remote patient monitoring and fitness trackers. • Agriculture: IoT helps farmers with things like crop health and animal tracking. • Transportation: IoT is used in cars and trucks for navigation and safety. • Energy Savings: IoT helps save energy in homes and manages electricity grids. • Environmental Monitoring: It's used to check air and water quality and protect wildlife. • Smart Cities: IoT manages traffic and waste in cities. • Building Control: It helps with things like lighting and heating in offices.	
D.5 Differentiate between the components of an ICT system	**Link to D.4, D,7 and C.7, R.3 – R.7**
Identify the key components of an IoT system: • Devices: Physical objects connected to the internet that can be used to collect and exchange data • Sensors: These collect data, e.g. temperature sensor, movement • Actuators: These perform actions based on data, e.g. a thermostat that adjusts the heat • Microcontrollers/Microprocessor: 	Reference to Rs.

Content (Grade 9 / Term 2)	Notes/Examples
D.6 Explain how the adaptation of technology impacted the world we work and live in	**Link to D,4, D.5 and R.1 – R.7**
Internet of Things link to Rs **Example Activity – Use KWLS chart to learn more about IoT** Provide learners with a KWLS chart on paper. Learners compete the first two columns (what I know and what I want to know), then learners watch the following video: https://youtu.be/6mBO2vqLv38?t=36 Learners then complete the last two columns of the KWLS chart while watching the video. Discuss the following example of an IoT device: **Example** For example, let's take a smart fridge. This fridge is part of the Internet of Things because it has sensors inside that can detect how cold it is, how full it is, and what items are inside. It can then send this information over the internet to your phone or computer. So, if you're at the grocery store and forgot to make a shopping list, you can check your phone to see what's inside your smart fridge. https://youtu.be/6mBO2vqLv38 https://youtu.be/_AlcRoqS65E [What is IoT? - A Simple Explanation of the Internet of Things (iotforall.com)](https://iotforall.com) https://bit.ly/3owhfZF  Devices/Things  Communication  Basic Components  IoT, Cloud & Big Data   Terms/Concepts	**D.5 and D.8 done together** **Link to Rs** Learners need to • Understand what the IoT is • Understand what the basic components of IoT is • Understand how it impacts our lives • Provide an example **Impact** IoT makes virtually everything "smart". The IoT is gradually becoming one of the most prominent ICT technologies that underpin our society. It presents a unique technology transition that is impacting all our lives. Some people refer to it as **Internet of Everything**. Link to microcontrollers (Rs) The Internet of Things (IoT) describes the network of physical objects - "things"—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet. These devices range from ordinary household objects to sophisticated industrial tools. **Refer to any object that is connected to the internet and can exchange information with other devices online.**
D.7 Present a basic understanding of the concept of input processing and output.	**Link to D.4, D.5, D.6 and Rs.**
Example activity - The process of how sensors measure data, transmit it to a microcontroller, and trigger an actuator Provide learners with the following diagram:  Sensors(measure the temperature) Continuosly measures the temperature Controller (decision maker) If temperature gets above certain temperature then turn on the fan Else no action Actuator (cooling fan) .  Revise and explain the concept of input, process, output concept with reference to IoT. • Sensors are devices that detect and measure physical properties or environmental conditions, such as temperature, light, pressure, or motion • Some sensors generate analogue signals in response to the physical property they are sensing. Analog signals are continuous and vary smoothly (e.g. turning the volume of a radio up or down). • As computing devices work with binary, analogue data needs to be converted or mapped to digital data	**Do in relation to Rs.** Also link to IPO tables Learners need to • understand input, processing and output with regards to microcontrollers • distinguish between digital and analogue in terms of input and output • describe the process involved in how sensors measure data, is sent to the microcontroller and executed by the actuator **Digital** uses discrete steps, like light switches (two states), to represent information, while **analogue** is like a continuous flow, such as turning up the volume on a radio. Both have their uses in the world of technology and can be found in various devices all around us. Think of a digital system as a series of tiny switches that can be either on or off, like a light switch. In the digital world, everything is represented using these on-off switches. Each

Content (Grade 9 / Term 2)	Notes/Examples
Measure data experiment Acceleration (Speed) project Using the accelerometer to obtain readings - Simple geared DC motor buggy. Micro: bit Application using accelerometer readings in an array (list) in short intervals. The microcontroller is attached to the buggy using Prestik or tape for the experiment.  Display the recorded readings.  on button B pressed show string join "Reading 1" list get value at 0 pause (ms) 500 show string join "Reading 2" list get value at 1 pause (ms) 500 show string join "Reading 3" list get value at 2  on start set list to array of 0 0 0 on button A pressed show icon play tone Middle C for 1 beat until done show leds pause (ms) 2000 list set value at 0 to acceleration (mg) x pause (ms) 1000 list set value at 1 to acceleration (mg) x pause (ms) 1000 list set value at 2 to acceleration (mg) x **Alternate experiment (Rocket)** Also see: https://makecode.microbit.org/courses/ucp-science/rocket-acceleration/overview	switch is like a tiny piece of information, called a "bit." When you put many bits together, you can represent more complex things, like numbers, letters, pictures, and sounds Analogue, on the other hand, is more like a smooth, continuous flow. Imagine a dimmer switch for a light instead of a simple on-off switch. With an analogue system, things can have any value within a range, and there are no discrete steps. It's like turning up the volume on a radio; you can have it at any level you want, not just loud or quiet.
D.8 Interpret a pattern to represent or communicate a message or image	**D.8 and D.9 done together.**
D.9 Create a pattern to represent or communicate a message or image	**Link to C.6 and C.7**
Example activity 1: - Convert ASCII code to binary. Explain what ASCII (American Standard Code for Information Interchange) is and its role in representing characters in computers. Mention that each character has a unique decimal value assigned to it. Show an example of ASCII art (simple drawings or patterns created using ASCII characters). Explain that the challenge is to decode ASCII art into binary. Divide the learners into teams or pairs, depending on the group size. Provide each team with a different piece of ASCII art and a corresponding list of ASCII characters and decimal values. Their task is to decode the ASCII art into binary by finding the decimal values for each character and converting them. They can use coloured markers or pencils to make the binary representation visually engaging. **Example activity 2**: - Extend the activity to create bingo cards with ASCII characters and their corresponding binary values. Call out ASCII characters, and learners mark the corresponding binary values on their Bingo cards.	Revise binary code.
D.10 Demonstrate a basic proficiency in the application of digital skills.	**Link to C.2 – C.5, and R.5 – R.7**
Learners develop a presentation (PP) about using/applying the use of an IoT device (Link to Ds above). Example: Live webcam e.g. Explore Africa Example connecting to and watching the feed on a IoT device: https://youtu.be/VUJbDTIYIM4	Link to networks (IoT)

3.3.3 Term 3

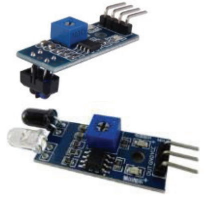
Content (Grade 9 / Term 3)	Notes/Examples
C.1 Apply computational thinking (CT) skills to develop a set of logical instructions to solve a problem.	**Link to C.2 – C.7 and R.5 – R.7**
C.2 Present a simple coding solution using symbolic or written statements representing sequences of commands, single repetition, and conditional constructs.	**C.1 and C.2 done together**
Example activity – Word scrambler and crack-the-word game (Lists, string handling) Pair programming activity. The purpose of this game is that one sprite inputs a word and then displays a scrambled version of the word. The other sprite then must guess what the word is. The sprite who displayed the scrambled word must say if the guess was right or wrong. The program must stop when the word is guessed correctly. Possible solution: • Determine the length of the original word. • Populate a list with the letters of the word (Use a For loop from 1 to the length of the word). • Choose a random letter from the list and join it to form another string. • When a letter was selected, delete it from the word (so that it cannot be chosen again). **Example activity Follow up on the word scrambler program.** Cooperative learning activity – group size of four or less. Explain how the "hangman" game works. Give step-by-step instructions as a first attempt to write a program for the game "hangman".	
C.3 Interpret and execute a given symbolic or written set of commands	
Example activity – Expand word scrambler by using the code below: Expand the program to do the following: • Add a counter to indicate how many guesses it took to guess the correct word. • Display how many letters of the word that the user guessed are in the correct position. 	**Note:** Generally, most learners are comfortable with tasks/problems that require them to write code that requires them to combine one or two concepts at a time. Manny learners, however, struggle when they must combine coding concepts at a time/in one program as it increases the difficulty level as well as the complexity of the task/problem. It is therefore advisable that leaners practise coding concepts using small, basic, manageable tasks/problems until they are ready for the next step.

```
when this sprite clicked
ask I will mix this word and you have to guess the word. and wait
set OriginalWord to answer
set Counter to 1
repeat length of OriginalWord
    add letter Counter of OriginalWord to WordList
    change Counter by 1

set MixedLetters to
repeat until length of WordList = 0
    set Counter to pick random 1 to length of WordList
    set MixedLetters to join MixedLetters item Counter of WordList
    delete Counter of WordList

when I receive NOPE
say Nope for 2 seconds

when I receive Yesss
say join Well done you found: OriginalWord for 2 seconds
```

Content (Grade 9 / Term 3)	Notes/Examples
C.4 Debug a given symbolic or written set of instructions	**Link to C.2, C.3**
Debug the age calculator The algorithm below must act as an age calculator. The algorithm was compiled to calculate the following: • Calculate how old you will be in the current year. • Calculate if you were born in a leap year • Convert your age in years to your age in days. • Convert your age in years to your age in hours. Do the following: • Compile a trace table to test if the age calculator works correctly and correct the algorithm where necessary. • Implement the program in Scratch and add the following code: ○ Add the code to convert your age in years to your age in seconds. ○ Add the code to display the season (summer, winter, etc.) in which you were born.	**Link to C1, C2, C3, C4, C5, C6 and C7**

```
~Age calculator
  Enter the year in which you were born
  Input iYear
  Enter the month in which you were born
  Input iMonth
  Enter the day in which you were born
  Input iDay
  Enter the current year
  Input iCurrentYear
  iAge <-- iYear – iCurrentYear    //Calculate your age in the current year
  If iYear mod 4 = 0 then //Calculate your age if you were born in a leap year
     Display “You were born in a leap year”
  Else
     Display “You were not born in a leap year”
  EndIf
  rAgeInDays <-- iAge*365.25     //Convert your age in years to your age in
days
  RAgeInHours <-- iAge * 24       //Convert your age in years to your age in
hours
```

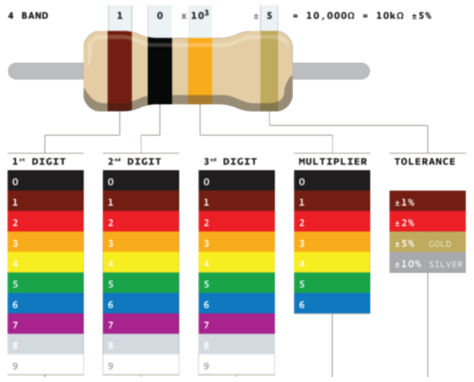
C.5 Evaluate a given solution towards potential improvement	**Link to C.2 – C.4**
Provide learners with code to evaluate	
C.6 Recognise and interpret patterns in symbolic sets of data or visualisations.	
C.7 Create or complete a pattern to represent a data set	
Example activity – Tribonacci numbers Tribonacci numbers work like Fibonacci numbers. The Tribonacci numbers start with three predetermined terms and each term afterwards is the sum of the preceding three terms. The Tribonacci numbers T_n are defined as follows: $T_1 = 0$, $T_2 = T_3 = 1$, and $T_n = T_{n-1} + T_{n-2} + T_{n-3}$ (n ≥ 3) In other words, each number is the sum of the previous three numbers. The first few Tribonacci numbers with signature (0,1,1) are: 0, 1, ,1, 2, 4, 7, 13, 24 … (**Note**: The first three terms are the signature. The signature may vary.) Use the experience with the Fibonacci problem, to develop the algorithm for coding the Tribonacci numbers	Learners use computational thinking, pattern recognition and previous experience of a similar problem to write an algorithm to display Tribonacci numbers.
Robotics	
R.3 Outline the different components of a robot	
Learners need to describe and use the following sensors	Learners are introduced to IR proximity sensor The learners need to provide: • A basic definition of a motor driver board. • Explain the purpose of the board. • Explain how it is used in robotics projects. **Infrared Obstacle Avoidance Sensors – Breadboard experiment**

Component	**Basic definition**	**Purpose**	**Practical application example – May be used to:**
IR proximity sensor 	An IR (Infrared) proximity sensor is a device that uses infrared light to detect the presence or absence of an object in its vicinity. It emits infrared radiation and measures the reflection or absorption of this radiation to determine the object's distance or proximity.	The primary purpose of an IR proximity sensor is to detect the presence or proximity of objects without physical contact. It is used to trigger specific actions or responses when an object enters or leaves a defined detection range.	In a robotics project, an IR proximity sensor can be utilized to enable obstacle avoidance. For instance, a robot equipped with IR proximity sensors can detect nearby obstacles as it moves and adjust its path or stop to avoid collisions. This enhances the robot's ability to navigate autonomously in a dynamic environment, making it suitable for applications such as autonomous vacuum cleaners, line-following robots, or drones that need to maintain a safe distance from obstacles during flight.

Content (Grade 9 / Term 3)

Resistor values

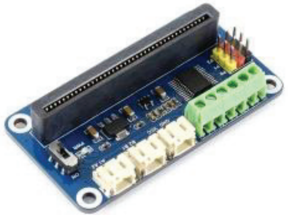
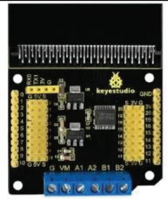
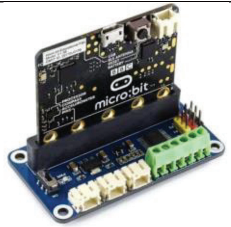
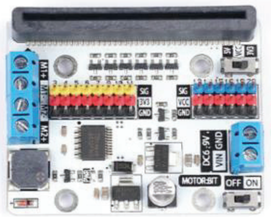
Revisit the concept of a resistor and its use in robotics and other electronics projects.
Calculate the resistance on a 4-band resistor.



Link activity CODING – the learners can develop a block-based program to show the resistance of a particular resistor if the four colour codes are supplied

The learners need to know the basic principles and use of a motor driver board as part of robotics and physical computing artefacts

Examples of motor / servo driver boards

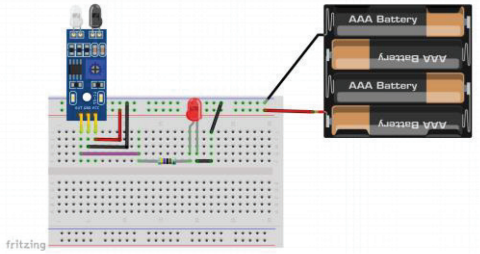
 Sample motor driver board	 Sample motor driver board	 Sample motor driver board	 Sample motor driver board

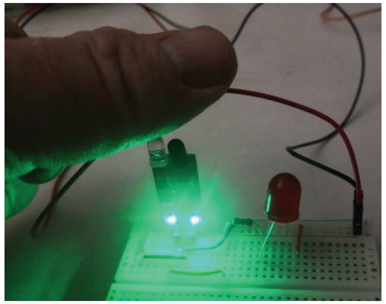
Component	Purpose	Usage in Robotics, in respect of a (Microcontroller) An example
A motor driver board for a microcontroller is a hardware component that interfaces with a microcontroller to control and drive electric motors. It provides the necessary circuitry and controls to manage the speed and direction of one or more motors.	The purpose of a motor driver board with a microcontroller is to enable the microcontroller to control motors, allowing it to perform tasks like driving wheels on a robot, spinning propellers on a drone, or moving other mechanical components. It simplifies motor control by handling the high-power requirements and providing an interface for the microcontroller to send commands to the motors.	In a robotics project involving a microcontroller a motor driver board can be used to drive the wheels of a robot. For instance, a small line-following robot can use two DC motors connected to the motor driver board. The microcontroller sends signals to the motor driver board to control the speed and direction of these motors, enabling the robot to move forward, backward, turn, or follow a line on the ground. This integration simplifies motor control and allows the microcontroller to focus on higher-level tasks like sensor readings and decision-making for the robot's behaviour.

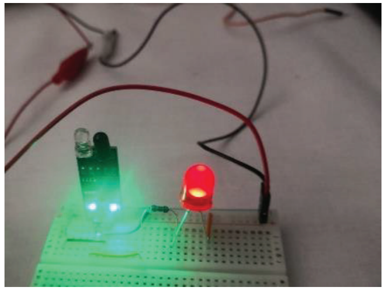
For enrichment purposes and for the highflyer learners (extended opportunity) an introduction to display components could be given.

Notes/Examples

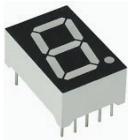
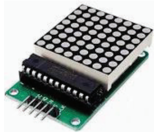
This is a simple breadboard project illustrating the basic operation of a sensor that presents a signal. It shows the basic principles of the operation of a sensor. The purpose of the experiment is to outline how the sensor could be used in an LFR.





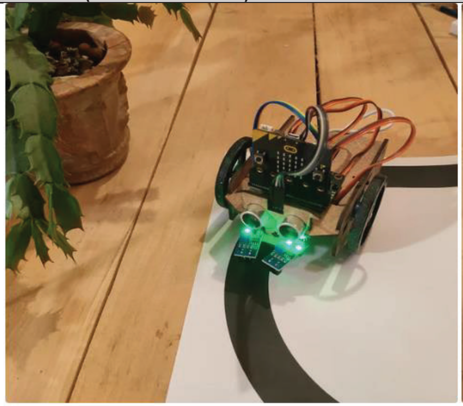
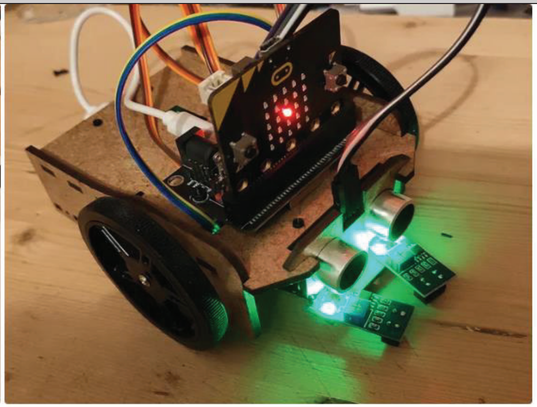


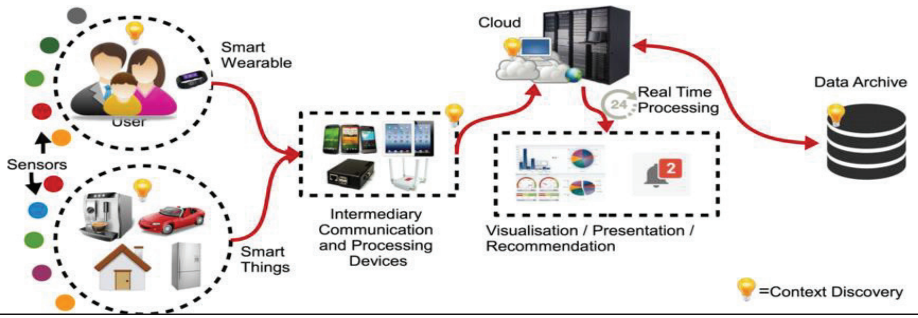
Content (Grade 9 / Term 3)	Notes/Examples

FOR ENRICHMENT	Basic Definition	Purpose	Usage in Robotics, in respect of a (Microcontroller) An example
7 Segment display 	A 7-segment display is a digital numeric display consisting of seven LED segments arranged in a specific pattern to represent digits from 0 to 9. It can display numerical values and some limited characters.	The primary purpose of a 7-segment display is to show numeric data in a simple and compact format. It's widely used in various applications where numerical information needs to be conveyed, such as digital clocks, thermometers, or countdown timers.	In a digital thermometer project, a 7-segment display can be utilized to exhibit the current temperature in a digital format. The individual segments can be lit up to represent the digits of the temperature value, providing a user-friendly and easily readable output
8 x 8 LED display  Or 4 In 1 (8x8(Display 	An 8x8 LED dot matrix display is an array of 64 individually controllable LED lights arranged in an 8x8 grid. Each LED can be independently turned on or off, allowing for the display of text, numbers, symbols, or animations. A 4 in 1, 8x8 LED display module combines four 8x8 LED dot matrix displays into a single unit, resulting in a larger display area. Each of the four displays can be controlled individually or together to create larger and more detailed visual outputs.	The purpose of an 8x8 LED dot matrix display is to provide a visually versatile output for information or graphics. It's commonly used for displaying text messages, simple graphics, or scrolling messages in various electronic projects. The purpose of a 4 in 1, 8x8 LED display module is to provide a more extensive and versatile LED display capability compared to a single 8x8 display. It's commonly used in projects where a larger or more impactful visual display is required, such as for scrolling messages, animations, or larger text.	In a digital clock project, an 8x8 LED dot matrix display can be employed to showcase the time in a visually appealing manner. By lighting specific LEDs to form numbers and colons, the display can continuously update to show the current time, giving a distinctive and dynamic appearance. In a digital scoreboard project for a sports event the display module can be employed to show scores and timing information. The larger display area allows for clear visibility of scores from a distance, and it can also be used to present animations or team logos.
16x2 LCD Display (with ic2 module) 	A 16x2 LCD (Liquid Crystal Display) is a character-based alphanumeric display that consists of 16 columns and 2 rows of characters. It's capable of displaying text and limited graphical symbols.	The purpose of a 16x2 LCD display is to provide a text-based output interface for displaying information in a clear and readable format. It's commonly used in embedded systems, devices, and electronic projects to convey important information.	In a weather station project, a 16x2 LCD display can be used to show real-time weather data such as temperature, humidity, and atmospheric pressure. The display can present this information in an organized manner, allowing users to easily access the current weather conditions.
Neopixel Strip 	A strip of individually addressable RGB (Red, Green, Blue) LEDs that can display a wide range of colours and patterns.	Used for dynamic and colourful lighting effects, visual displays, and feedback in robotics projects.	Neopixel strips are employed for aesthetic purposes, indicating status, or enhancing the visual appeal of a robot. They can also be used for signalling and communication.

The learners need to know the elementary difference between digital and analogue input and how to process the readings in block-based code.

R.5 Design a simple artefact based on a set of design instructions	**Link to R.6 – R.7 and C.1 – C.7 and D.6 and D.7**
R.6 Mimic the operations of a robot	R.5, R.6 and R.7 done together
R.7 Create, test and execute a set of robotic instructions	
Example of an LFR and OAR, implementing IR Sensors and an Ultrasonic sensor. Example from (Dame, A. 2021 - DIY Microcontroller Educational Mobile Robot V2 [Used with permission] See (https://www.instructables.com/DIY-Microbit-Educational-Mobile-Robot-V2/)	Learners must investigate and describe the typical composition of a Line following robot using IR sensors, applying the principles of design thinking. The learners should be able to describe the use and purpose of the following components typically used as part of a Line following robot (LFR) (on a conceptual level only) • Chassis

Content (Grade 9 / Term 3)	Notes/Examples
   **IMPORTANT** Schools should always consider the affordability of components and kits.	● Wheels ● Power supply ● Microcontroller ● Motors and ● Motor driver ● Breadboard and jumper wires for prototyping ● IR Sensor/s The learners should also be able to describe the logical principles guiding the code of an LFR.
Digital Concepts	
D.1 Outline the concept of technology and purpose of information technology (IT)	**D.1, D.3, D.4, D.5 done together**
D.2 Recognise that he or she is living as citizens in a digital world.	**D.2, D.6, D.8, D.9 done together**
Relate to IoT. **Phishing** is a fraudulent attempt to obtain sensitive information, such as usernames, passwords, and financial details, by disguising it as a trustworthy entity. **Ransomware** is malicious software that encrypts a user's files or system and demands a ransom in exchange for decryption keys. Learners watch the following videos: https://youtu.be/XsOWczwRVuc and https://youtu.be/xbHCrgAl2n4 **Example activity**: Describe a scenario where a city's smart grid, controlled by IoT devices, has been attacked by ransomware. The devices are locked, and the city's power supply is at risk. Provide participants with a puzzle using binary code. They must solve it to unlock the devices and save the city's power grid. Answer the following question in binary. 01001000 01100101 01101100 01101100 01101111 00101100 00100000 01010100 01101000 01101001 01110011 00100000 01101001 01110011 00100000 01100001 00100000 01100010 01101001 01101110 01100001 01110010 01111001 00100000 01100011 01101111 01100100 01100101 00100000 01110000 01110101 01111010 01111010 01101100 01100101 00101100 00100000 01100001 01110010 01100101 00100000 01111001 01101111 01110101 00100000 01110011 01101111 01101100 01110110 01101001 01101110 01100111 00111111 Solution: "Hello, this is a binary code puzzle, are you solving?" The solution to the puzzle is: "Yes, I am solving." 01011001 01100101 01110011 00101100 00100000 01001001 00100000 01100001 01101101 00100000 01110011 01101111 01101100 01110110 01101001 01101110 01100111	Learners need to understand ● What malware is and what it includes, ● Describe Phishing ● Describe Ransomware
D.3 Demonstrate an understanding of the concept of a computing device.	**D.1, D.3, D.4, D.5 done together**
Case Study: IoT in Smart Home Automation Smart home automation involves connecting household devices and systems to the internet, allowing homeowners to control them remotely through smartphones, tablets, or voice commands. This technology enhances convenience, security, energy efficiency, and overall comfort. Meet the Johnson family. They live in a suburban home and want to make their living space more efficient and comfortable. To achieve this, they decide to implement IoT-based smart home automation solutions. **Smart Thermostat**: The Johnsons install a smart thermostat that learns their preferences and adjusts the temperature accordingly. They can control the thermostat remotely, ensuring their home is comfortable when they arrive. **Smart Lighting**: IoT-enabled light bulbs are installed throughout the house. The family can control the lighting using their smartphones or set schedules for lights to turn on and off, enhancing security when they're away.	

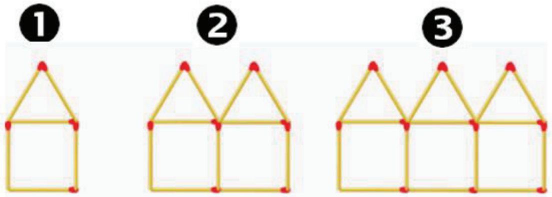
Content (Grade 9 / Term 3)	Notes/Examples
Smart Security System: The Johnsons set up a smart security system that includes cameras, motion sensors, and door/window sensors. They receive real-time alerts on their devices if any suspicious activity is detected, allowing them to take immediate action. **Smart Appliances**: The family upgrades to smart appliances like a refrigerator with a built-in camera. This allows them to see the contents of the fridge while grocery shopping and receive expiration date notifications. **Voice Assistants**: The Johnsons use voice assistants like Amazon Echo or Google Home to control various devices using voice commands. They can ask about the weather, set reminders, and control other IoT devices. **Benefits:** **Convenience**: The Johnsons can control their home environment remotely, adjusting settings even when they're not at home. **Energy Efficiency**: The smart thermostat and lighting system help save energy by optimizing usage based on occupancy and preferences. **Security**: Real-time monitoring and alerts enhance the security of their home, deterring potential intruders. **Resource Management**: The family can monitor energy and water usage, helping them make informed decisions about conservation. **Challenges and Considerations:** **Privacy Concerns**: Connected devices gather data about users' habits and preferences, raising concerns about data privacy. **Compatibility**: Not all smart devices are compatible with each other, making it important to choose a cohesive ecosystem. **Initial Costs**: Setting up a smart home requires an investment in devices and infrastructure. The Johnsons' experience with their smart home illustrates how IoT can significantly improve the quality of life and bring convenience to daily routines. The case study showcases how smart home automation, driven by IoT technology, offers homeowners the ability to control their environment, enhance security, and promote energy efficiency. As IoT continues to evolve, the concept of the smart home is likely to become more integrated into our lives, making our living spaces more comfortable and connected. Some questions for discussion: 1. What is the concept of a "smart home" in this case study, and how does IoT play a role in it? 2. Why do you think smart home automation is becoming increasingly popular? 3. Explain the functionality and benefits of the smart thermostat installed by the Johnsons. 4. How do IoT-enabled light bulbs contribute to both convenience and security in their home? 5. Describe the components and functions of the smart security system used by the Johnsons. 6. What advantages do the Johnsons gain from upgrading to smart appliances like the refrigerator with a built-in camera? 7. How do voice assistants enhance the Johnsons' control over their smart home devices? 8. Discuss the convenience aspects of the Johnsons' smart home. How does remote control improve their daily life? 9. How do the smart thermostat and lighting system help the Johnsons save energy? What role does optimization play? 10. Explain how the smart security system not only enhances security but also provides peace of mind. 11. Analyse the privacy concerns related to IoT devices in smart homes. How can users address these concerns? 12. Can you think of other real-life scenarios where IoT technology could be applied to enhance convenience, security, or energy efficiency?	
D.4 Identify the common uses of ICT in the real world	• **Link to D.1 and D.5**
Understand how networks support the Internet of Things (IoT) and cloud computing. **Example activity**: Provide learners with a blank diagram of the IoT and cloud network architecture. Ask them to label and draw arrows to represent how data flows from an IoT device to the cloud and back. Encourage them to include labels for IoT devices, local networks, the internet, and cloud servers. E.g. 	**Revise networks** **Revise IoT**

<table>
<tr><th>Content (Grade 9 / Term 3)</th><th>Notes/Examples</th></tr>
<tr><td>D.5 Differentiate between the components of an ICT system</td><td>D.4, D.5, D.7 done together</td></tr>
<tr><td>D.6 Explain how the adaptation of technology impacted the world we work and live in</td><td>D.2, D.6, D.8, D.9 done together</td></tr>
<tr><td>D.7 Present a basic understanding of the concept of input processing and output.</td><td>Link to R.5 – R.7</td></tr>
<tr><td>Scenario: Smart Home Climate Control System
Divide learners in pairs and let them create an IPO table to plot the different processes.
Input:
In a modern smart home, the climate control system is equipped with various sensors, including temperature sensors, humidity sensors, and motion sensors. The residents of the home have set their preferred temperature range to be between 22°C and 25°C and a humidity level of around 40-50%. The system also considers the residents' occupancy patterns, such as when they are at home and when they are away.
Process:
The smart home climate control system's microprocessor constantly receives input from the sensors, which monitor the temperature, humidity, and detect motion in different rooms of the house. The microprocessor analyses this data to determine the current climate conditions within the home.
Output:
Based on the processed information, the smart home climate control system takes appropriate actions to maintain the desired climate conditions. Here are some examples of how the system responds:
If the temperature is below the lower threshold (22°C) and there are occupants in the room, the microprocessor activates the heating system to increase the temperature.
If the temperature exceeds the upper threshold (25°C) and occupants are present, the microprocessor activates the air conditioning system to lower the temperature.
If the humidity level falls below 40%, the microprocessor turns on a humidifier to increase humidity.
When the motion sensors detect that there are no occupants in the house, the microprocessor adjusts the climate control settings to an energy-saving mode, maintaining a slightly wider temperature range to conserve energy.
If no occupants are detected for an extended period, the system goes into a sleep mode, reducing power consumption until someone returns.
By continually processing the input from various sensors, the microprocessor ensures that the smart home's climate control system operates efficiently, providing a comfortable living environment while optimizing energy usage based on occupancy patterns.</td><td>
<table>
<tr><th>Input</th><th>Processing</th><th>Output</th></tr>
<tr><td>Temperature sensors</td><td>Microprocessor collects temperature data</td><td>Analyses indoor temperature for climate control</td></tr>
<tr><td>Humidity sensors</td><td>Microprocessor measures humidity levels</td><td>Evaluates humidity for climate regulation</td></tr>
<tr><td>Motion sensors</td><td>Microprocessor detects occupancy status</td><td>Adjusts climate settings based on occupancy</td></tr>
<tr><td>User-defined preferences</td><td>Microprocessor uses residents' settings</td><td>Adapts climate settings to match user preferences</td></tr>
<tr><td>Time of day data</td><td>Microprocessor tracks day and night periods</td><td>Optimizes climate control based on time of day</td></tr>
<tr><td>Other environmental data (optional)</td><td>Microprocessor processes additional data</td><td>Customizes climate actions based on specific needs or environmental factors</td></tr>
</table>
The IPO table outlines the inputs, processing, and outputs of the Smart Home Climate System. The microprocessor gathers data from temperature sensors, humidity sensors, motion sensors, user-defined preferences, and time-of-day. It then processes this information to analyse indoor temperature, humidity levels, occupancy status, and time of day. Based on this processing, the system adjusts climate settings to regulate temperature and humidity levels, for a comfortable living environment. It considers user-defined preferences for personalized comfort. The system can customize climate actions based on specific needs or environmental factors to optimize energy usage and promote energy-efficient practices.</td></tr>
<tr><td>D.8 Interpret a pattern to represent or communicate a message or image</td><td>D.2, D.6, D.8, D.9 done together</td></tr>
<tr><td>D.9 Create a pattern to represent or communicate a message or image</td><td>D.2, D.6, D.8, D.9 done together</td></tr>
<tr><td>D.10 Demonstrate a basic proficiency in the application of digital skills.</td><td>Link to C.2 – C.5 and R.5 – R.7</td></tr>
<tr><td>Combine with coding activities.</td><td></td></tr>
</table>

3.3.4 Term 4

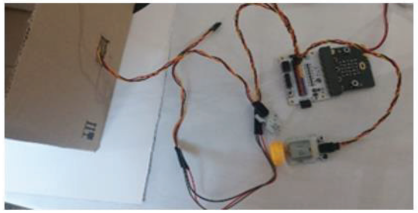
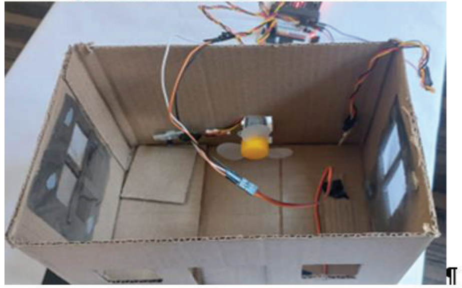
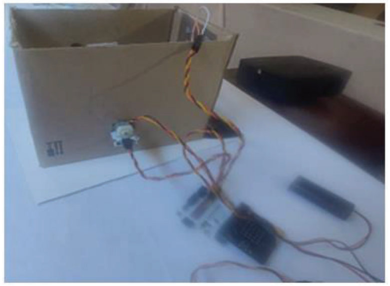
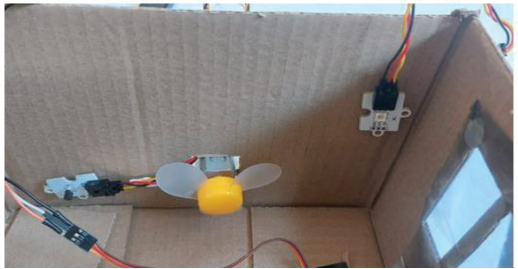
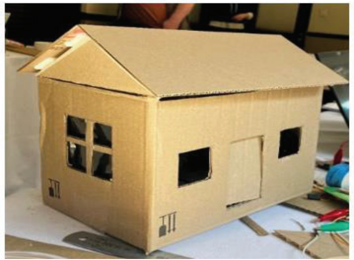
<table>
<tr><th>Content (Grade 9 / Term 4)</th><th>Notes/Examples</th></tr>
<tr><td>Coding</td><td></td></tr>
<tr><td>C.1 Apply computational thinking (CT) skills to develop a set of logical instructions to solve a problem.</td><td>Link to C.1 – C.7 and R.5 – R.7</td></tr>
<tr><td>C.2 Present a simple coding solution using symbolic or written statements representing sequences of commands, single repetition, and conditional constructs.</td><td rowspan="5">C.1, C.2, C.3, C.4 and C.5 are done together</td></tr>
<tr><td>C.3 Interpret and execute a given symbolic or written set of commands</td></tr>
<tr><td>C.4 Debug a given symbolic or written set of instructions</td></tr>
<tr><td>C.5 Evaluate a given solution towards potential improvement</td></tr>
<tr><td>Example activity – Create a game
Watch the following videos for ideas and use your experience from the games you wrote in previous terms to create a game of your choice
Snakes & ladders boardgame:
Creating dice https://youtu.be/qt3wjftisry
Creating the board https://youtu.be/-aqt9jnyaok
Playing the game https://youtu.be/1l3c0y4l6ou
Maze Game https://youtu.be/zEH1JOIbD5o
Racing Game https://youtu.be/aYnKI0fEXA4</td></tr>
<tr><td>C.6 Recognise and interpret patterns in symbolic sets of data or visualisations.</td><td>Link to D.6 and D.7 and C.1</td></tr>
<tr><td>C.7 Create or complete a pattern to represent a data set</td><td rowspan="2">C.6 and C.7 are done together as the activity require recognising and interpreting as well as completing a pattern

Note:
Concrete activities remain important as literature suggests that the primary weakness of today's pedagogy of programming is that it does not provide enough opportunity for the novice to develop concrete operational skills, via the correct types of exercises…due to too much emphasis on writing large amounts of code, and problem solving.

By identifying patterns, we can predict what will come next and what will happen again and again in the same way.
In Computer Science/coding we analyse patterns in data and make predictions and generalisations based on the pattern analysis.</td></tr>
<tr><td>Example activity
Study the picture on the right. Each house pattern is built using a certain number of match sticks.
1) Complete the following table for pattern no 4, 7 and no 10
<table>
<tr><td>No of houses</td><td>1</td><td>2</td><td>3</td><td>4</td><td>...</td><td>7</td><td>...</td><td>10</td></tr>
<tr><td>Number of match sticks used</td><td>6</td><td>11</td><td>16</td><td></td><td></td><td></td><td></td><td></td></tr>
</table>


2) From the table above, Identify and describe the pattern
3) Look for a relationship between the number of match sticks and the number of houses, e.g. to build the single house (pattern no 1), you need 6 match sticks and for the two houses (pattern no 2) you need 11 match sticks, etc.
Complete the table and describe the pattern between the pattern no and the number of match sticks, e.g. 1 and 5; 2 and 10, etc
<table>
<tr><td>No of houses</td><td>1</td><td>2</td><td>3</td><td>4</td><td>...</td><td>7</td><td>...</td><td>10</td></tr>
<tr><td>Number of match sticks used</td><td>5 + 1</td><td>10 + 1</td><td>15 + 1</td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
+5 +5

If n is the pattern no, use the relationship from above to generalise the pattern and represent the generalised pattern in symbolic format (using n) for each of the pattern numbers on the right
<table>
<tr><td>No of houses</td><td>1</td><td>2</td><td>3</td><td>4</td><td>...</td><td>n</td><td>150</td></tr>
<tr><td>Number of match sticks used</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table></td></tr>
</table>

Content (Grade 9 / Term 4)	**Notes/Examples**
Robotics	
R.3 Outline the different components of a robot	R.**3, R.5, R.6 and R.7 done together**
R.5 Design a simple artefact based on a set of design instructions **R.6 Mimic the operations of a robot** **R.7 Create, test and execute a set of robotic instructions** **Final Project**  **Example Project – Smart Home** The learners must construct their artefact from materials and smart home from recycled material.      	The learners must be able to apply the principles of design thinking to develop an artefact that deploys at least 2 sensors, an actuator, and other components. The learner may also decide to incorporate any additional sensors, or controls and communication with another microcontroller. The learner should present the overall design of their artefact, its purpose and how it is to operate. The theme for the project is a SMART device. Examples of such projects are: - a smart home, or - smart farm/hydroponic system, - or even a smart robot (E.g., Balloon popper etc.)
R.4 – Present an understanding of how robots affect the world.	
Example Activity The learners can brainstorm on how the development of creative solutions to address concerns related to robot-human interactions, employment shifts, and privacy concerns, promoting responsible and ethical implementation could be addressed in the 4IR+ society	The learners should be able to • Explain the possibilities of advanced robots with artificial intelligence, considering their impact on employment, social structures, and human interaction. • Discuss the advantages and disadvantages of utilizing robots in dangerous situations and space exploration.
Digital Concepts	

Content (Grade 9 / Term 4)	Notes/Examples
D.1 Outline the concept of technology and purpose of information technology (IT)	
D.2 Recognise that he or she is living as citizens in a digital world.	**Link to D.6**
D.3 Demonstrate an understanding of the concept of a computing device.	**Link to D.1, D.4 and D.5**
D.4 Identify the common uses of ICT in the real world	**Link to D.1 and D.5**
D.5 Differentiate between the components of an ICT system	**Link to D.4**
D.6 Explain how the adaptation of technology impacted the world we work and live in	**Link to D.2**
D.7 Present a basic understanding of the concept of input processing and output.	**Link to D.3, C.1-C.5 and R.5-R.7**
D.8 Interpret a pattern to represent or communicate a message or image.	
D.9 Create a pattern to represent or communicate a message or image.	
D.10 Demonstrate a basic proficiency in the application of digital skills.	**Link to C.2 – C.5 and R.5 – R.7** Consolidate all outcomes into one or two activities.
Consolidation activity: Create a visual representation of your ideal smart home by sketching or using digital design tools. Include rooms, devices, connections, and designated roles for family members or role players within your smart home. Be sure to consider the layout, placement of devices, and how you interact to make your home more efficient and enjoyable. Learners will need a piece of paper, pencils, markers, and any other materials they want to use to create their visual representation. This activity can be done in groups of 3 – 4. Steps to follow: **Brainstorm ideas**: Brainstorm a list of features they would like their ideal smart home to have. This could include things like: • Smart lights that can be turned on and off with a voice command or a gesture. • A security system that can be monitored remotely. • A thermostat that can be adjusted automatically to save energy. • A doorbell that can send a notification to your phone when someone is at the door. • A robot vacuum cleaner that can clean the floors without you having to lift a finger. • A smart lock that can be unlocked with your fingerprint. • A smart speaker that can play music, set alarms, and answer questions. • A smart TV that can stream movies and shows, control other devices, and even make video calls **Draw a floor plan**: Draw a floor plan of their ideal smart home. Label each room and indicate where the different devices would be located. **Add rules for use**: Add rules for how the different devices in your smart home would be used. For example, you might decide that the lights can only be turned on and off by adults, or that the security system can only be activated when someone is home. **Connect the devices**: Indicate how the different devices in your smart home would be connected to each other. This could be done with wires, Wi-Fi, or Bluetooth. **Label the devices**: Label each device in their smart home with its name and function. **Describe the communication between devices**: Describe how the different devices in your smart home would communicate with each other. For example, the lights would automatically turn on when the security system is activated, or the thermostat would adjust itself to save energy when no one is home. **Describe the security features:** Describe the security features that would be in your smart home. This could include things like motion sensors, door and window sensors, and security cameras. **Describe the use for each role player**: Describe how each person in the household would use the different devices in the smart home. For example, parents would use the security system to keep their children safe, or the children would use the robot vacuum cleaner to help them with their chores. Once learners have completed this activity, they will create a visual representation of their ideal smart home. They will also have a better understanding of how smart homes work and how they can be used to make life more convenient and efficient. Additional tips for this activity: Encourage learners to be creative and think outside the box. There are no right or wrong answers when it comes to designing a smart home. Help learners to research different smart home devices and features. This will give them a better understanding of the options available to them. Encourage learners to collaborate with each other. This will help them to come up with more creative and innovative ideas. Allow learners to revise their designs as needed. This is a learning process, and it is important to give learners the opportunity to make mistakes and learn from them. The activity can be extended to include a garden system.	

4 SECTION 4
ASSESSMENT

4.1 ASSESSMENT

Assessment is a continuous planned process of identifying, gathering and interpreting information about the performance of learners, using various forms of assessment. It involves four steps: generating and collecting evidence of achievement, evaluating this evidence, recording the findings and using this information to understand and thereby assist the learner's development to improve the process of learning and teaching.

Assessment involves activities that are undertaken throughout the year. Assessment comprises two different but related activities: informal daily assessment (assessment for learning) and formal assessment (assessment of learning).

Assessment in Coding and Robotics should encourage computational thinking practices, i.e. integrating the power of human thinking with the capabilities of ICTs and computer programming.

However, cognisance should also be taken of what is being assessed. Certain competencies are best assessed with particular forms of assessment. Different kinds of assessments are appropriate to the competencies necessary for different topics at different age groups. It is useful to use an observation checklist to assess learners measuring in the early grades. Rubrics, for example, can be used to evaluate learner's Coding and Robotics as well as problem solving skills.

Assessment involves activities that are undertaken throughout the year. In grades 7 – 9 assessment comprises two different but related activities: informal daily assessment (assessment for learning) and formal assessment (assessment of learning).

Assessment in Coding and Robotics should encourage computational thinking practices, i.e. integrating the power of human thinking with the capabilities of ICTs and computer programming.

4.1.1 Informal or daily assessment

Assessment for learning has the purpose of continuously collecting information on a learner's achievement that can be used to improve their learning. Informal assessment is the daily monitoring of learners' daily progression and should also focus on how learners learn and retain new information. It should therefore include retrieval practice (as described by the science of learning –section 2.7.4) as well as deliberate practise (See Section 2.7.5).

Trying to remember something enhances memory, and teachers can use quizzes or self-tests for this purpose. As learners learn and retain new information by focusing on the meaning of the content, teachers can assign tasks that require learners to explain or organise the material (e.g. concept maps), which helps them think about the meaning of content.

In learning Coding and Robotics, practise is also essential, and teachers can focus on regular practise and retrieval as well as spaced practise and retrieval over time to aid long-term retention. Teachers can also interleave different types of practice and use multiple modalities to enhance learning

Informal assessment and retrieval practise may be as simple as stopping during the lesson to ask questions or have learners writing down what they can remember about what was learned in a previous lesson and provide feedback to the learners. Informal assessment does not need be recorded. It is part of all learning activities taking place in the classroom. Learners or teachers can mark these tasks.

Self-assessment and peer assessment actively involves learners in assessment. This is important as it allows learners to learn from and reflect on their own performance. The results of the informal daily assessment tasks are not formally recorded unless the teacher wishes to do so. The results of daily assessment tasks are not used for promotion and certification purposes.

4.1.2 Formal assessment

All assessment tasks that make up a formal programme of assessment for the year are regarded as formal assessment. Formal assessment tasks are marked and formally recorded by the teacher for progression and certification purposes. All formal assessment tasks are subject to moderation for the purpose of quality assurance and to ensure that appropriate standards are maintained.

Formal assessment provides teachers with a systematic way of evaluating how well learners are progressing in a grade and in a particular subject. Examples of formal assessments include tests, examinations, practical tasks, projects, etc. Formal assessment tasks form part of a year-long formal programme of assessment in each grade and subject.

Coding and Robotics has three strands with vital practical components regarding the Coding and Robotics strands.

Both the Coding and Robotics strands include mini–Practical Assessment Tasks (PATs) which together will make up 33,3% of the Final Examination mark (20 out of a possible 60).

The following tables provide the formal assessment requirements for Coding and Robotics:

Table 4-12 Minimum formal assessment requirements for Coding and Robotics

	Informal Assessment	**Formal Assessment per Term**		
		Practical Tasks and Theory Test/Examination		**Total**
	Enabling Activities/Tasks	**Practical Tasks**	**Term Test/Examination**	**Term Mark**
Term 1	0%	70% (1 Coding + 1 Robotics – 35+35)	30%	100%
Term 2		70% (1 Coding + 1 Robotics – 35+35)	30%	100%
Term 3		70% (1 Coding + 1 Robotics – 35+35)	30%	100%
Term 4		70 marks = 100%	No Test	100%
Promotion Mark	**CASS Component: 40%**	**Final Examination Component: 60%**		
	Continuous Assessment: Test + Practical Tasks: 40	**Combined Practical Tasks: 20**	**Examination: 40**	**Promotion**
	Term 1 + Term 2+ Term 3 + Term 4	T1 + T2 + T3 + T4	**40**	**100**
	10 + 10 + 10 + 10	**5 + 5 + 5 + 5**		

Note:

The practical tasks must include practical content/application with regards to

- Coding (e.g. develop code for a coding task/problem – refer to C.2)
- Robotics (e.g. creating robotics artefacts – refer to R.5 – R.7)

Practical tasks for robotics could be done and assessed in pairs.

The tests and examination must include theoretical content with regards to

- Coding (e.g. writing an algorithm, correcting/debugging or evaluating a coding algorithm/solution, explaining what code does, interpreting/creating patterns, etc.)
- Robotics (e.g. What a robot is, types of robots, impact of robots, etc.)
- Digital concepts

The forms of assessment used should be age and developmental level appropriate. The design of these tasks should cover the content of the subject and include a variety of tasks designed to achieve the objectives of the subject.

4.2 PROBLEM-BASED LEARNING

Assessment in Coding and Robotics can be done assessing the learner in action, for example, watching the learner solving the problem without stopping the moment. This can be done using the following strategies:

4.2.1 Individual Problem-based Learning (coding)

Problem solving is the process of designing, evaluating, and implementing a strategy to answer question, complete a task or achieve a desired goal.

4.2.1.1 Types of problems

In terms of coding, typically, problems could require learners to

- provide missing code instructions (code instructions are provided with some instructions or code elements missing / to be completed or
- choose the correct solution from 2-3 options or
- work through (trace) / act out code to determine if it is correct and correct if required or
- rewrite a set of coding instructions to be more efficient or
- compare different solutions to evaluate efficiency or
- translate verbal/written instructions to code (e.g. packing arrows)
- develop the solution/algorithm (code instructions) themselves using computational thinking and following problem-solving process.

The above will depend on the competency the learner needs to demonstrate. Coding problems need to gradually increase in terms of complexity.

4.2.1.2 Assessing problem-based learning (coding)

The learner is assigned a problem he/she must solve and in doing so

- needs to understand the problem.
- analyses the problem (what is given and what is needed / what is important and what can be ignored - abstraction).
- identifies the main steps (abstraction / high level solution).
- identifies the detailed steps (decomposition / breaking down the main steps).
- Identifies patterns to determine the need for using coding structures such as repetition.
- implements and tests the solution (algorithm).
- debugs the solution if required.

Refer to Annexure A for rubric example to assess problem solving.

4.2.2 Cooperative Learning

Instead of encouraging learners to compete for grades or achievement, cooperative learning asks them to work together and participate in group learning activities (small groups, e.g. 4 learners), under the guidance of a teacher.

Assessing cooperative learning in Senior Phase Coding and Robotics

Example rubric to assess cooperative learning activity: *Defining a robot and its different parts*.

Refer to Section 2.6.2.1 for example cooperative learning activity.

Refer to Annexure A for rubric example to assess cooperative learning.

4.2.3 Pair Programming

Assessing pair programming in Senior Phase Coding and Robotics

Example rubric to assess cooperative learning activity:

Identifying, completing and creating patterns.

Refer to Section 2.6.2.2 for example pair programming learning activity.

Refer to Annexure A for rubric example to assess pair programming.

4.3 RECORDING AND REPORTING

Recording is a process in which the teacher documents the level of a learner's performance in a specific assessment task. It indicates learner progress towards the achievement of the knowledge as prescribed in the Curriculum and Assessment Policy Statements. Records of learner performance should provide evidence of the learner's conceptual progression within a grade and her / his readiness to progress or being promoted to the next grade. Records of learner performance should also be used to verify the progress made by teachers and learners in the teaching and learning process.

Reporting is a process of communicating learner performance to learners, parents, schools, and other stakeholders. Learner performance can be reported in several ways. These include report cards, parents' meetings, school visitation days, parent-teacher conferences, phone calls, letters, class or school newsletters, etc. Teachers in all grades report in percentages against the subject. The various achievement levels and their corresponding percentage bands are as shown in the Table below.

RATING CODE	DESCRIPTION OF COMPETENCE	PERCENTAGE
7	Outstanding achievement	80 – 100
6	Meritorious achievement	70 – 79
5	Substantial achievement	60 – 69
4	Adequate achievement	50 – 59
3	Moderate achievement	40 – 49
2	Elementary achievement	30 – 39
1	Not achieved	0 - 29

4.4 GENERAL

This document should be read in conjunction with:

- National policy pertaining to the programme and promotion requirements of national Curriculum statement Grades R-12; and
- The policy document, National Protocol for Assessment Grades R-12

ANNEXURE A: TERMINOLOGY

The following tables provide clarity on terminology used in the Coding and Robotics CAPS

A.1 CODING

Table A-13 Coding - Clarification of concepts and terms

Term/Concept	Explanation
Algorithm	An algorithm is a set of logical instructions/commands that a human or computer can execute to solve a specific problem or accomplish a particular task. It is a computational process that uses a finite number of steps (logical instructions or commands), carried out in a specific sequence to solve a problem.
Coding	**Coding** is the process of writing instructions that a computer can understand and execute These instructions are written in a programming language, which is a set of rules that define how the instructions should be written. The purpose of coding is to create software programs that can perform specific tasks, such as running a website, playing a video game, or analysing data.
Computation	In computing, **computation** refers to any type of arithmetic or non-arithmetic calculation that is well-defined. It can involve mathematical equations, computer algorithms, and other types of calculations.
Computational Thinking	It refers to a problem-solving approach that involves breaking down complex problems into smaller, more manageable parts and using algorithms and logical reasoning to solve them. It involves skills such as abstraction, decomposition, pattern recognition, and algorithmic thinking. It is a way of thinking that is used in computer science, but it can also be applied to other fields. In education, computational thinking is used to teach learners how to think logically and solve problems systematically.
Conditional (choice/ decision) statement	A control structure that selects one alternative from two or more possible execution sequences to be executed
Control statement	A control structure that is used to modify the order in which instructions are executed such as a loop or conditional statement
Event	A signal or notification that something has happened.
Expression	Refers to a combination of one or more values, operators that can be evaluated to produce a result.
Input	In computing, input refers to the data that is entered into a computer system, such as text, images, or sound,
IPO table	Input-Processing-Output table describes the inputs processing and outputs of program.
Loop statement	A control structure that allows a sequence of instructions to be continually repeated until a certain condition is reached
Operator	**Operators** are symbols or keywords that represent computations or actions performed on operands. Operators include: Arithmetic operators (+, -, x, /, modulo), comparison operators (=, >, <, ≤, ≥, ≠), Boolean operators OR, AND, NOT, string operators for manipulating strings/text (length, concatenate, indexing) Operators provide the building blocks for creating expressions and performing operations
Output	In computing, output refers to the result of the processed data that is presented to the user in a usable format. This can be in the form of text, sound, image, or video.
Processing	In computing, processing refers to the operations performed by the computer to manipulate or analyse the input data.
Program	A **program** is a sequence of instructions that a computer can execute to perform a specific task.
Trace table	In programming, a trace table is a technique used to test an algorithm and predict step by step how the computer will run the algorithm. Statements are executed step by step, and the values of variables change as an assignment statement is executed. A trace table simulates the flow of execution by showing the values of variables at each step of the algorithm. Trace tables are typically used by novice programmers to understand how an algorithm works and to identify errors in the algorithm 2
Variable	In programming, a variable is a named storage location that holds a value or data. Variables are essential for storing and manipulating data in computer programs. The values in variables can change during the execution of a program.

A.2 ROBOTICS

Table A-14 Robotics - Clarification of concepts and terms

Term/Concept	Explanation
Actuator	Refers to a device that converts energy into physical motion, such as rotation or translation. Actuators are often called the muscles of robots, as they enable robots to perform various tasks and interact with the environment
Binary Code	Refers to a system of representing data using only two symbols, typically 0 and 1. It is the foundation of digital technology and is used in computers and other digital devices to store, process, and transmit data and information.
Boolean Logic	Refers to is a branch of mathematics and logic that deals with the values of true and false, usually denoted by 1 and 0. Boolean logic uses operators such as AND, OR, and NOT to combine and manipulate these values, and to create expressions that represent logical conditions.
Capacitor	Refers to a type of electronic component that can store electric charge or energy when voltage is applied across it.
Controller	Refers to a device that commands, directs, and regulates the behaviour of a robotic system. It takes input signals from the robot's sensors, processes them based on programmed instructions, and then sends output signals to the robot's actuators to perform the desired actions.
Logic Gates	Refers to devices that perform logical operations on one or more binary inputs and produce a single binary output. They are the basic building blocks of digital circuits and systems, such as computers, calculators, and phones.
Microcontroller	Refer to a type of small computer that can control the functions and behaviour of a robotic system. It generally consists of a processor, memory, input/output ports and other peripherals that can be programmed to perform specific tasks. It can receive data from sensors, process it according to the programmed instructions and send commands to actuators.
Robot	A **robot** is a machine that can perform a series of actions automatically, either by being programmed by a computer or by being guided by an external control device.
Sensor	Refers to a device that can measure or detect some physical property of the environment or the robot itself and convert it into an electrical signal. Examples include light sensor, touch sensor, sound sensor, etc.
Servo	Refers to a type of motor that can rotate or move to a specific position, angle, or speed with high accuracy and precision.

A.3 DIGITAL CONCEPTS

Table A-15 Digital Concepts - Clarification of concepts and terms

Term/Concept	Explanation
Cipher	A cipher, also known as an encryption algorithm, is a set of well-defined rules used to transform information into a scrambled form, called ciphertext. It is used to encrypt messages so that they can only be read by someone who knows how to decrypt them.
Cloud	Refers to the use of remote servers over the internet to store, process, and access data and applications.
Cloud Computing	Refers to the on-demand availability of computing resources, such as storage, infrastructure, software, and data, over the internet.
Computing device	A general-purpose machine that can execute instructions for any data processing purpose. A computing device can receive input, do something with the input and provide a result or output.
Cybercrime	It refers to any illegal activity that involves or uses a computer, a computer network, or a networked device. Cybercrime can target individuals, corporations, or governments, and can cause various types of harm, such as financial loss, identity theft, privacy violation, or data damage.
Data	Raw, unprocessed facts and figures.
Decode	Reconstructing the original **(encoded)** information. It involves taking an encoded representation and converting it back into its original form
Decrypt	The reverse process of encryption, taking ciphertext and using the appropriate key to convert it back into its original, readable plaintext form.
Digital Citizen	A person who uses the Internet and other digital technology to communicate with other and engage in society.
Digital Citizenship	The ability to participate in online society.

	It includes concepts like respecting others' privacy, avoiding cyberbullying, netiquette, digital health and welfare, ability to assess the credibility and reliability of online information, intellectual property, impact and responsibility of online actions and deeds.
Digital Footprint	The trail of traceable digital activities, actions, contributions, and communications one leaves behind when using the Internet or digital devices.
Encode	Converting information into a specific format (transforming data or messages into another format)
Encrypt	The process of transforming readable data (plaintext) into an unreadable, scrambled form (ciphertext) using a cryptographic algorithm (cipher) and a secret key.
Hardware	The physical building blocks of a computing device or the tangible parts you can see and touch. It includes: • **Central Processing Unit** (CPU): the component responsible for executing instructions. • **Random Access Memory** (RAM): Component for temporary storage of programs and data the computing device is currently working with. • **Storage devices**: E.g. hard drives, solid-state drives (SSDs), for permanent data storage. • **Input devices** such as keyboard, mouse, screen, microphone mouse, used to interact with the computer. • **Output devices** such as screen, speakers, printer, etc., used to display and output information.
Information	Data that has been processed and organised to convey meaning.
Information and Communications Technology (ICT)	ICT is the use of computing and telecommunication technologies, systems, and tools to facilitate the way information is created, collected, processed, transmitted, accessed and stored
Information Technology (IT)	IT refers to the use of computer systems to manage, process, protect, and exchange data and information.
Input	In computing, input refers to the data that is entered into a computer system, such as text, images, or sound.
Internet	Refers to a global network of interconnected computers and other devices that use standardised communication protocols to exchange data and information. It allows billions of users worldwide to access and share a vast array of resources, including websites, documents, images, videos, and more.
Internet of Things (IoT)	It is the term for the network of physical devices, such as sensors, cameras, machines, and appliances, that can collect and share data over the internet or other communication networks. IoT devices can be controlled remotely, monitored in real time, and programmed to perform various tasks. IoT can be used for many purposes, such as smart homes, smart agriculture, smart healthcare, smart manufacturing, and smart transportation.
Output	In computing, output refers to the result of the processed data that is presented to the user in a usable format. This can be in the form of text, sound, image, or video.
Personal information	In computing, **personal information or personal data** is any information that can identify a person, from one’s name and address to one’s device identifier and account number.
Processing	In computing, processing refers to the operations performed by the computer to manipulate or analyse the input data. This includes executing software applications, performing calculations, sorting and filtering data, and running programs.
Software	The intangible programs and applications (instructions) that give life to the physical components. Examples include: • **Operating System** (OS) that manages the hardware resources and provides a platform for running other programs. (e.g., Windows, macOS, Linux) • **Application software**: Specific programs designed for performing tasks like word processing, image editing, games, etc. • **Programming languages** used to create new software by writing instructions the computer can understand.
Technology	Encompasses any tool, technique, or process used to solve problems and manipulate our environment. Technology is designed with a purpose of solving problems that meet human needs and wants. It refers to tools, machines, or devices that make our lives easier or better.
World Wide Web (WWW)	Refers to an information system that enables content sharing over the Internet through user-friendly ways. It allows documents and other web resources to be accessed over the Internet according to specific rules

ANNEXURE B: ASSESSMENT EXAMPLES

B.1 COOPERATIVE LEARNING

Example rubric to assess cooperative learning activity: *Defining a robot and its different parts (See section 2.6.1)*.

	Learner name	#Definition of robot	#Flashcards used well.	#Drawing illustrates robot	*Learner fulfilled role well
1.					
2.					
3.					
4.					

#Replace with suitable criteria depending on the task/problem
*Will remain the same irrespective of task/problem

Note:
Although all learners in the group get the same mark for the first three criteria, each learner gets an individual mark for the "Learner fulfilled role well" – this is based on how well each learner contributed based on their set role.

The teacher can give mark these while learners are completing the activity and hence it should not require much extra time.

Each of the aspects listed in the table above, could be assessed using the following example:

Aspect assessed	Beginning (1) 😑	Developing (2) 🙂	Accomplished (3) 😊	Exemplary (4) 🤩
Definition of concept, e.g. robot	Key information is missing (e.g. no parts included) and the definition is unclear and difficult to follow	Some key information is included, and the definition is fairly clear and easy to follow but may be incomplete not well organised.	Most of the key information is included (e.g. most of the parts) and it is mostly well-organised and easy to follow	The learner demonstrates full understanding in that the definition is well-organised, complete, and easy to follow.
Flashcard utilised well	Flashcards are not used effectively	Some attempt is made to use the flashcard to explain the concept, but it lacks detail and key information	Flashcards are used appropriately to explain the concept and includes most of the key information	Flashcards used effectively/innovatively to support a complete explanation of the concept and all key information
Drawing illustrates concept, e.g. robot	Drawing attempts to convey the concept, but the drawing is incomplete and/or difficult to interpret	Drawing includes some relevant details that may not all be accurate and conveys the concept but lack detail	Drawing includes most of the relevant and accurate details that appropriately convey the concept	drawing includes rich, and accurate details that effectively convey the concept.
Learner fulfilled role well	Learner does not understand his/her role and makes no contribution or unrelated contributions	Shares ideas or tries to fulfil her/his role, but does not work with group and most of the contributions are unrelated	Tries to understand his/her role and mostly makes relevant contributions. Can work on her/his part and take part in the group	Generates ideas and builds upon other's ideas to develop a larger plan. Works independently to do his/her part and is invested in the other group members (e.g. helps when needed, cares about the group product)

B.2 PAIR PROGRAMMING /COMPLETING A TASK IN PAIRS

Example rubric to assess pair programming activity: *Identifying, completing and creating patterns*.

	Learner name	**#Concept1**	**#Concept2**	**#Concept2**	***Learner fulfilled role well**
1.					
2.					

#Replace with suitable criteria depending on the task/problem
*Will remain the same irrespective of task/problem

Note:

Although both learners get the same mark for the first three criteria, each learner gets an individual mark for the “Learner fulfilled role well” – this is based on how well each learner contributed based on their set role.

The teacher can assign certain marks while learners are completing the activity and hence it should not require much extra time.

Each of the aspects listed in the table above, must be assessed using a rubric:

B.3 DESIGN THINKING

A process that emphasizes creativity, experimentation, and iteration to arrive at the best solution that meets user needs.				
Competencies	**Beginning (1)** 😑	**Developing (2)** 🙂	**Accomplished (3)** 😊	**Exemplary (4)** 🤩
Inspiration: **Learner applies creative thinking to create a product or complete a task**	• Demonstrates limited creative thinking and understanding of the problem or task	• Applies creative thinking to understand the problem or task and identifies some opportunities for innovation	• Applies creative thinking effectively to gain a deeper understanding of the problem or task and identifies significant opportunities for innovation.	• Demonstrates exceptional creative thinking and in-depth understanding of the problem or task, uncovering unique insights and opportunities for innovation
Ideation: **Learner can create own ideas to create a product or completing a task.**	• Unsure about what is expected so any idea is scattered or unfocused and ideas do not clearly connect to the problem or task.	• Generally, mimics ideas from others (rather than creating new ideas) that are related to the problem or task.	• Creates new ideas that include enough detail and that are directly related to the problem or task.	• Creates many clear ideas by considering lots of possibilities that focuses on key information and fully addresses the problem or task
Implementation: **Learner can use best ideas to create a product or complete a task.**	• Creates a product or performance, but the product has limited functionality or detail and does not clearly address the problem, or the product is not useful.	• Creates a product or performance with some functionality that is somehow related to the challenge or problem.	• Uses ideas to create a product or performance with good functionality that is directly related to the problem or task.	• Creates clear ideas to create a product or performance with precision and full functionality and that fully addresses the problem or task.
Testing & Improving	• Provides minimal or no feedback and does not reflect on the quality to consider improvements or iterations	• Collects some feedback and reflects somewhat on the quality for considering minor improvements or iterations	• Collects thorough feedback, reflects accurately on the quality to inform improvements, and iterates on the solution	• Collects extensive feedback, conducts rigorous testing, and iterates on the design or solution based on feedback, leading to transformative improvements.

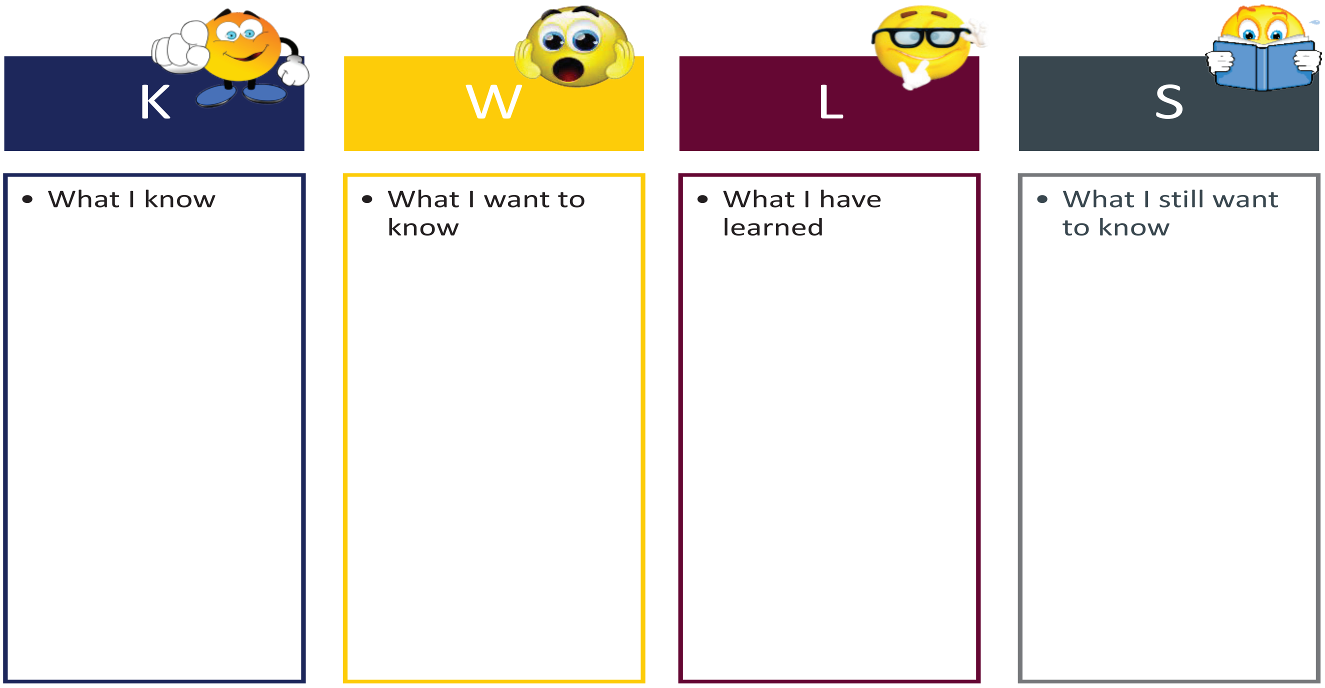
Note: All rubrics serve as examples only and may be adapted

ANNEXURE C: TEACHING RESOURCES

C.1 KWLS CHART

The KWLS chart is a learning strategy that helps learners engage with a topic in a structured and reflective manner. The chart helps learners organize their thoughts and track their progress as they explore a particular topic or concept.

The KWLS chart is a valuable tool for learners of all ages and levels of education. It promotes active engagement with the learning material, fosters critical thinking and inquiry, and supports metacognitive skills development. By using the KWLS chart, learners become more self-directed and aware of their learning process, leading to a more enriched and effective learning experience.



K	W	L	S
• What I know	• What I want to know	• What I have learned	• What I still want to know

K - What I Know: In this section, learners write down what they already know about the topic. This step helps them activate their prior knowledge and make connections with the new information they are about to encounter. Identifying what they already know also helps learners build a foundation for further learning and enables them to understand how the new information fits into their existing knowledge framework.

W - What I Want to Know: In this part, learners jot down questions or areas of interest they have about the topic. These are the aspects they hope to learn more about or understand better as they engage with the subject matter. This step encourages curiosity and sets the stage for active exploration. By noting down their questions, learners become more focused and motivated to seek answers and engage with the learning materials more critically.

L - What I Have Learned: As learners progress through their learning journey, they record the new information, insights, and understanding gained about the topic. This section allows learners to summarise and consolidate their learning experiences. It reinforces the concepts they have grasped and helps them reflect on the new knowledge acquired. Reflecting on what has been learned enhances comprehension and retention of the material.

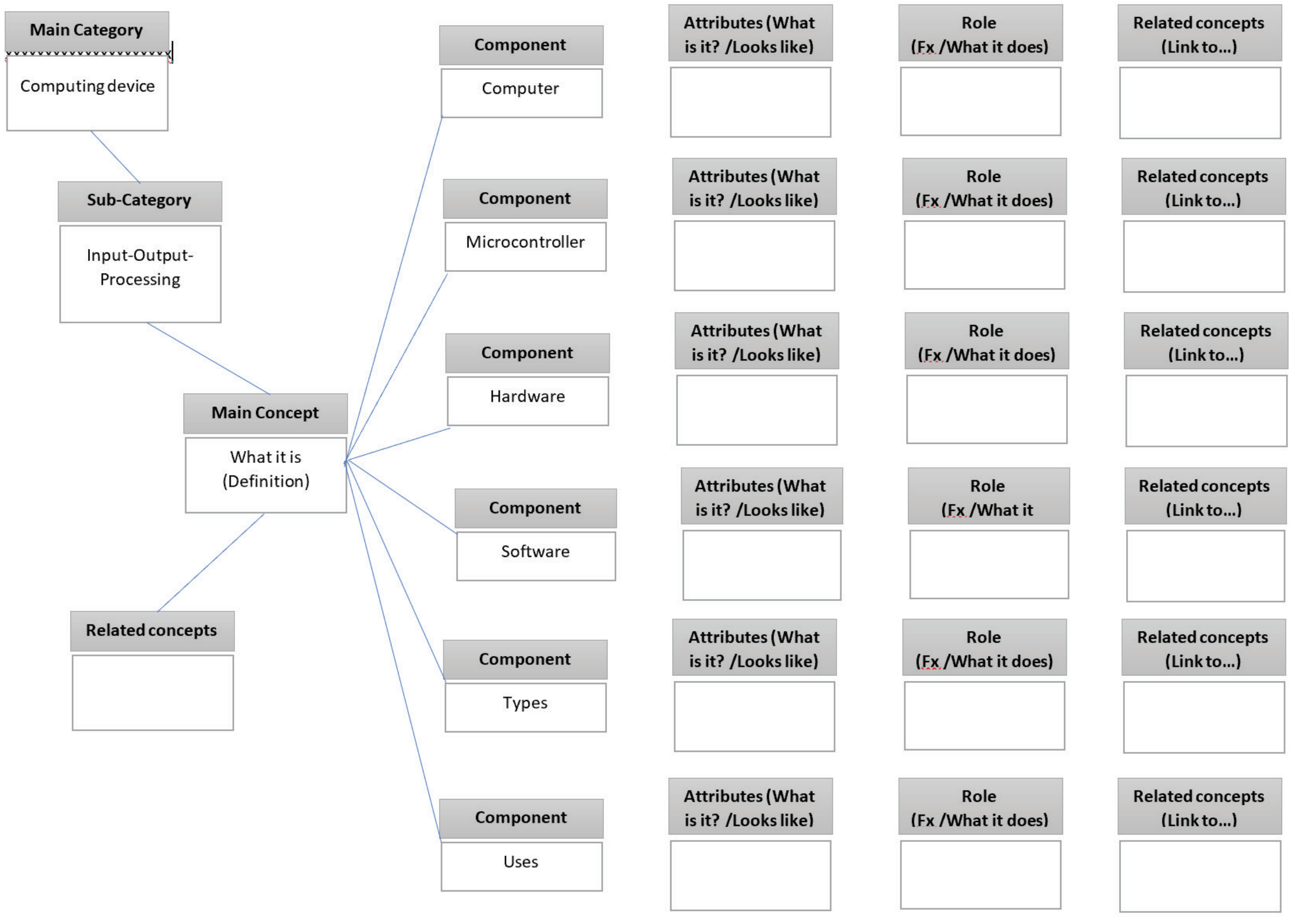
S - What I Still Want to Learn: In the last section, learners identify any remaining questions, uncertainties, or areas they would like to explore further. Even after learning a considerable amount about the topic, learners may realize that certain aspects still require clarification or deeper investigation. This step encourages a growth mindset, as learners recognize that learning is an ongoing process, and there is always more to discover.

C.2 CONCEPT MAPS

A concept map is a diagram that shows the relationships between different ideas. This helps you understand how they are connected. Every concept map — whether it is simple or complex — is made up of two key elements:

- **Concepts:** These are typically represented by circles, ovals, or boxes and are called "nodes."
- **Relationships:** These are represented by arrows that connect the concepts, and the arrows often include a connecting word or verb (but they do not have to). These arrows are called "cross-links."

Example of a simple high-level concept map for understanding computing devices



Other resources to be considered:

https://www.teachwithict.com/
http://code-it.co.uk/gold/
https://sites.google.com/gshare.blackgold.ca/blackgoldmicrobit/microbit [Microcontroller resource]
https://www.instructables.com/
https://www.101computing.net/bbc-microbit-counter-using-a-7-segment-display/
https://www.robotique.tech/type/microbit/