basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

CURRICULUM AND ASSESSMENT POLICY STATEMENT

(CAPS)

INFORMATION TECHNOLOGY

(March 2018)

Section 1

National Curriculum and Assessment Policy Statement for Information Technology

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contemplated in subparagraphs (a) and (b) comprise the following policy documents which will be incrementally repealed by the *National Curriculum Statement Grades R – 12 (January 2012)* during the period 2012-2014: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

(a) 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.

(b) 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.

(c) 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.

(d) The National Curriculum Statement Grades R - 12 aims to produce learners that are able to:

- 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.

(e) 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	**6**	**6**	**7**
▪ Beginning Knowledge	(1)	(1)	(2)
• Creative Arts	(2)	(2)	(2)
• Physical Education	(2)	(2)	(2)
• Personal and Social Well-being	(1)	(1)	(1)
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

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

SUBJECT	HOURS
Home Language	6
First Additional Language	5
Mathematics	6
Natural Science and Technology	3,5
Social Sciences	3
Life Skills	**4**
▪ Creative Arts	(1,5)
▪ Physical Education	(1)
▪ Personal and Social Well-being	(1,5)
TOTAL	**27,5**

1.4.3 Senior Phase

- The instructional time in the Senior Phase is as follows:

SUBJECT	HOURS
Home Language	5
First Additional Language	4
Mathematics	4,5
Natural Science	3
Social Sciences	3
Technology	2
Economic Management Sciences	2
Life Orientation	2
Arts and Culture	2
TOTAL	**27,5**

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.

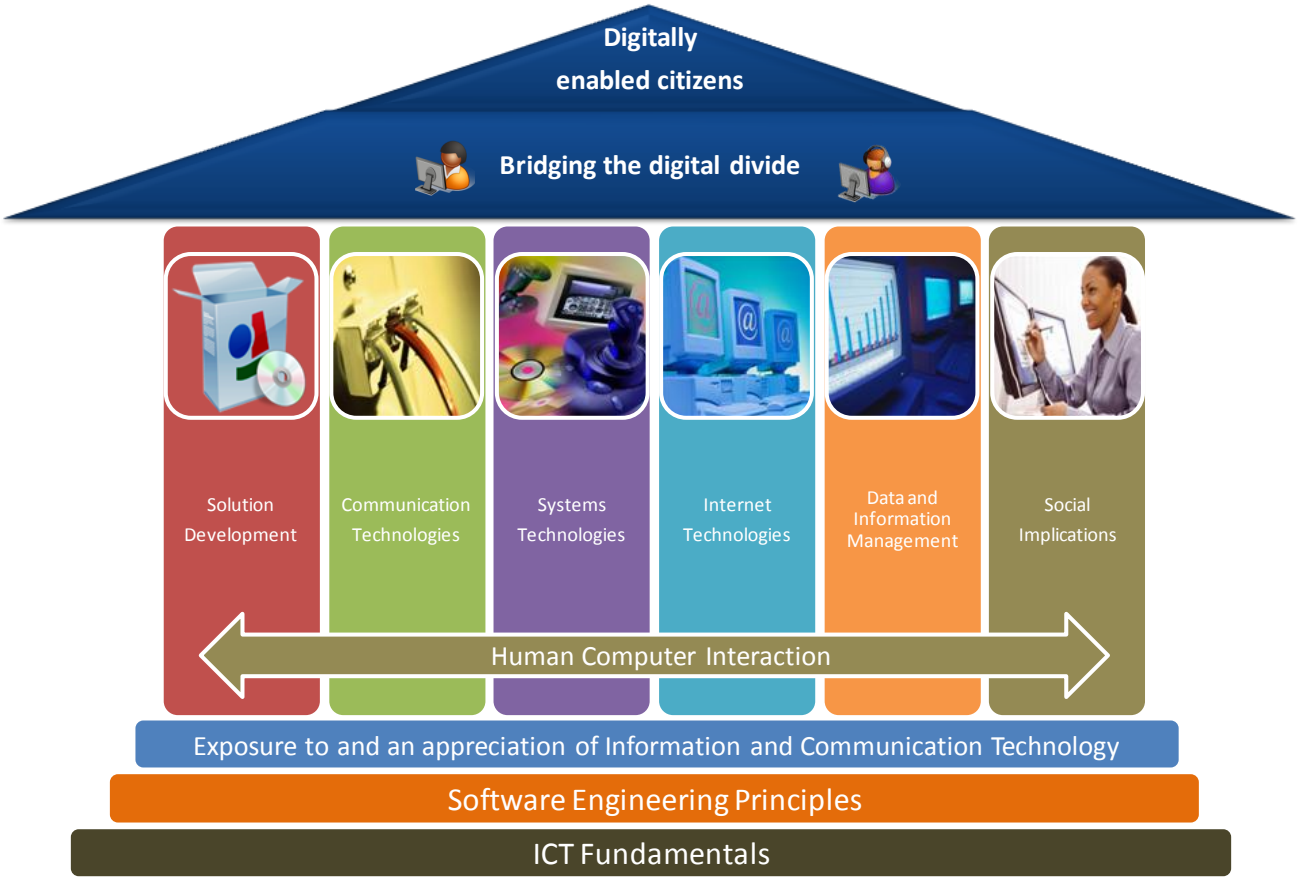
Section 2

Information Technology

2.1 What is Information Technology?

Information Technology is the study of the various interrelated physical and non-physical technologies used for the capturing of data, the processing of data into useful information and the management, presentation and dissemination of data. Information Technology studies the activities that deal with the solution of problems through logical and computational thinking. It includes the physical and non-physical components for the electronic transmission, access, and manipulation of data and information.

The diagram below illustrates how the six main topic areas of the Information Technology curriculum support the teaching of digitally informed learners.



The table below provides the six topics and sub-topics to be covered in Information Technology in grades 10 – 12 and the resources required for teaching IT:

Topic Area	Sub-Topics	Weighting (Content)	Resources
Solution Development	Algorithms and Problem Solving Introduction to Solution Development Application Development Software Engineering Principles	±60%	Computers Textbook Software • Database Management Software • High-level programming language within a visual development environment using an IDE with a GUI builder • Internet • Browser
Communication Technologies	Networks E-communication	±7%	
Systems Technologies	Introduction to Computers Hardware Software Computer Management	±10%	
Internet Technologies	Internet World Wide Web Internet Services	±8%	
Data and Information Management	Data Representation Database Management Database Design	±10%	
Social Implications	Legal Issues Ethical Issues Social Issues Environmental Issues Health Issues Computers and Society	±5%	

Topic links and overlap

It is important to note that there will always be a degree of overlap between topics. Solution development is enabled by systems technologies in the form of application software. Systems technologies allow for electronic communication. Electronic communication technologies enable the Internet, which is used for various applications that include information dissemination and electronic data interchange. Data and information management is a key concept and secondary activity overlapping concepts in many other areas such as solution development and Internet technologies. Data and information management is enabled by systems technologies. All ICT activities are primarily driven by human involvement, need and intervention, which in turn give rise to social and ethical issues.

For example, when teaching Communication Technologies, one could incorporate the social implications involved. This is also applicable to the Systems Technologies topic where the relevant social implications could be highlighted.

Approach

The curriculum is designed to introduce learners to the breadth of the field of Information Technology.

2.2 Specific aims of Information Technology

In Information Technology a learner will:

- use appropriate techniques and procedures to plan solutions and devise algorithms to solve problems using suitable techniques and tools'
- understand and use appropriate communication technologies for information dissemination;
- appreciate and comprehend the various systems technologies used in the developing of a computer-based system;
- understand that all ICT systems are built upon software engineering principles;
- understand and use Internet technologies for various tasks;
- comprehend and apply the concepts of data and information management to understand how a knowledge-driven society functions; and
- understand the social implications of ICTs and how to use ICT technologies responsibly.

2.3 Time allocation of Information Technology in the curriculum

In Grades 10 and 11 the time allocation for IT is 4 hours per week for 35 weeks. 5 weeks of the school year are taken up by examinations.

The Grade 12 time allocation is 4 hours per week for 28 weeks; 12 weeks of the school year are for examinations.

The table below provides suggestions for the ***approximate*** teaching time per topic:

	Grade 10		Grade 11		Grade 12	
Topic	**Hours**	**Weeks**	**Hours**	**Weeks**	**Hours**	**Weeks**
Solution Development	92	23	90	22.5	68	17
Communication Technologies	4	1	8	2	4	1
Systems Technologies	16	4	10	2.5	10	2.5
Internet Technologies	14	3.5	6	1.5	4	1
Data and Information Management	8	2	18	4.5	8	2
Social Implications	6	1.5	8	2	6	1.5
Teaching Time: Total	**140**	**35**	**140**	**35**	**100**	**25**
Examinations	**20**	**5**	**20**	**5**	**48**	**12**
TOTAL:	**160**	**40**	**160**	**40**	**148**	**37**

2.4 Resources required for offering Information Technology

Refer to circular S7 of 2006 for details on the resource requirements for the teaching of IT in Grades 10 – 12.

Infrastructure, equipment and finances for the subject are the responsibility of the school.

In Information Technology learners are required to work individually on a computer during contact time and need access to the Internet.

Schools should have a business plan for the subject that addresses the following:

- Initial capital layout for setting up a computer laboratory. The layout should provide for the following:
    - Entry-level computers (to ensure a lifespan of 4 – 5 years), networked
        - One computer per learner per period (during contact time)
        - Provision for sufficient computers to enable the practical examination to be completed in **two sittings**
    - One high-speed printer per computer room
    - Internet access
    - Data projector or demonstrating software
    - Software (operating system, Office suite, security software – antivirus, Internet security, software for solution development)
- Budget
    - Annual running costs
        - Software licensing (operating system, application software, security software, solution development software)
        - Cartridges, paper and storage media
        - Breakage and maintenance (regular service plan)
        - Insurance
        - Internet connectivity
    - Sustainability plan
        - To upgrade or replace software and equipment every 4 – 5 years.

Requirements for high-level programming tool to be used for software development:

High-level software development tool that includes an integrated development environment which:

- supports both structured and object oriented methodologies;
- uses a visual development environment with a graphical user interface builder; and
- allows for event-driven programming.

The GUI builder should allow for component based development with a WYSIWIG (what you see is what you get) editor utilising an event-driven architecture.

The development tool could also include software design utilities to facilitate the application of software engineering practices.

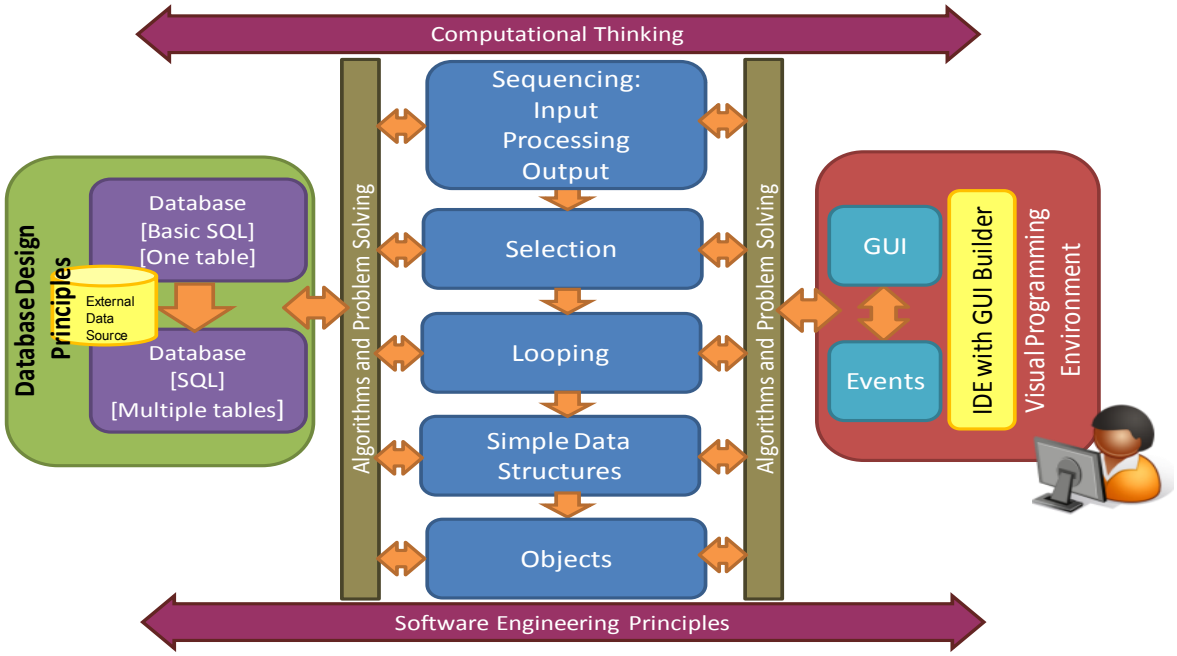
Section 3

Content and scope per topic

3.1 Solution Development

Solution development is the development of software in a planned and structured process and is based on solving computational problems which include data-related problems through logical thinking. It involves the practices of algorithm development and creating a software solution according to a set of rules and/or requirements specified in the problem statement or by a client/business/individual. The software is developed using appropriate problem-solving techniques, tools and methodologies. Software solution development is achieved through computer programming which could be based on a single or combination of development paradigms such as event-driven programming, object-oriented programming and sequential programming.

Broad topic layout and progression



Note:

In Grade 10 and 11 learners are introduced to important computational skills and concepts, algorithm development, problem solving and programming using a high-level programming language that uses an integrated development environment with a GUI builder. Learners are introduced to controls and code and basic object oriented programming (OOP). Event handling principles are reinforced using the form class, attributes, methods and controls.

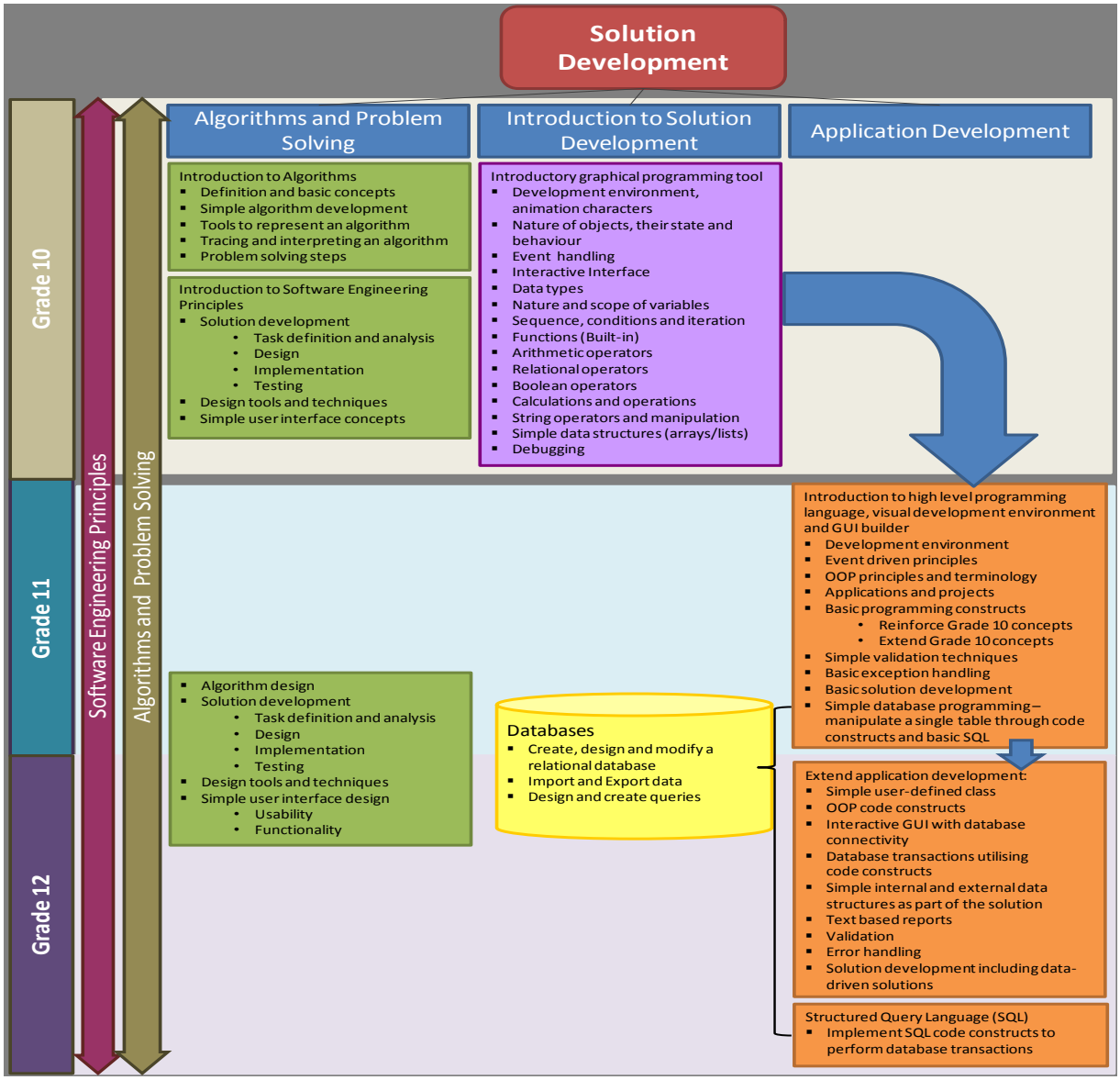
Skills to manipulate a database through code constructs are also introduced in Grade 11.

In Grade 12, the principles and constructs are further emphasised through more advanced concepts and problems and learners should be ready to engage with basic structured query language (SQL) code and manipulating a relational database.

The development of computational thinking practices of algorithm development, problem solving and programming underpin solution development and should be emphasised from Grade 10 to Grade 12.

Usability, HCI (human computer interaction) and software engineering principles should be reinforced as part of software development as well as when dealing with websites as part of the Internet Technologies topic.

Sub-topic layout and progression for Solution Development



Note:

Algorithmic problem solving in Grade 10 should be dealt with separately at first as an introduction to solution development to develop the learner's computational thinking practices of algorithm development, problem solving and programming using everyday scenarios.

Learners should develop an understanding of the importance of order and precision when developing an algorithm as well as the place of algorithms in software solutions and computing science. Thereafter it should be reinforced, extended and integrated with solution development and programming.

Solution development includes computational thinking and the application of software engineering principles using event-driven programming within the object-oriented (OO) paradigm.

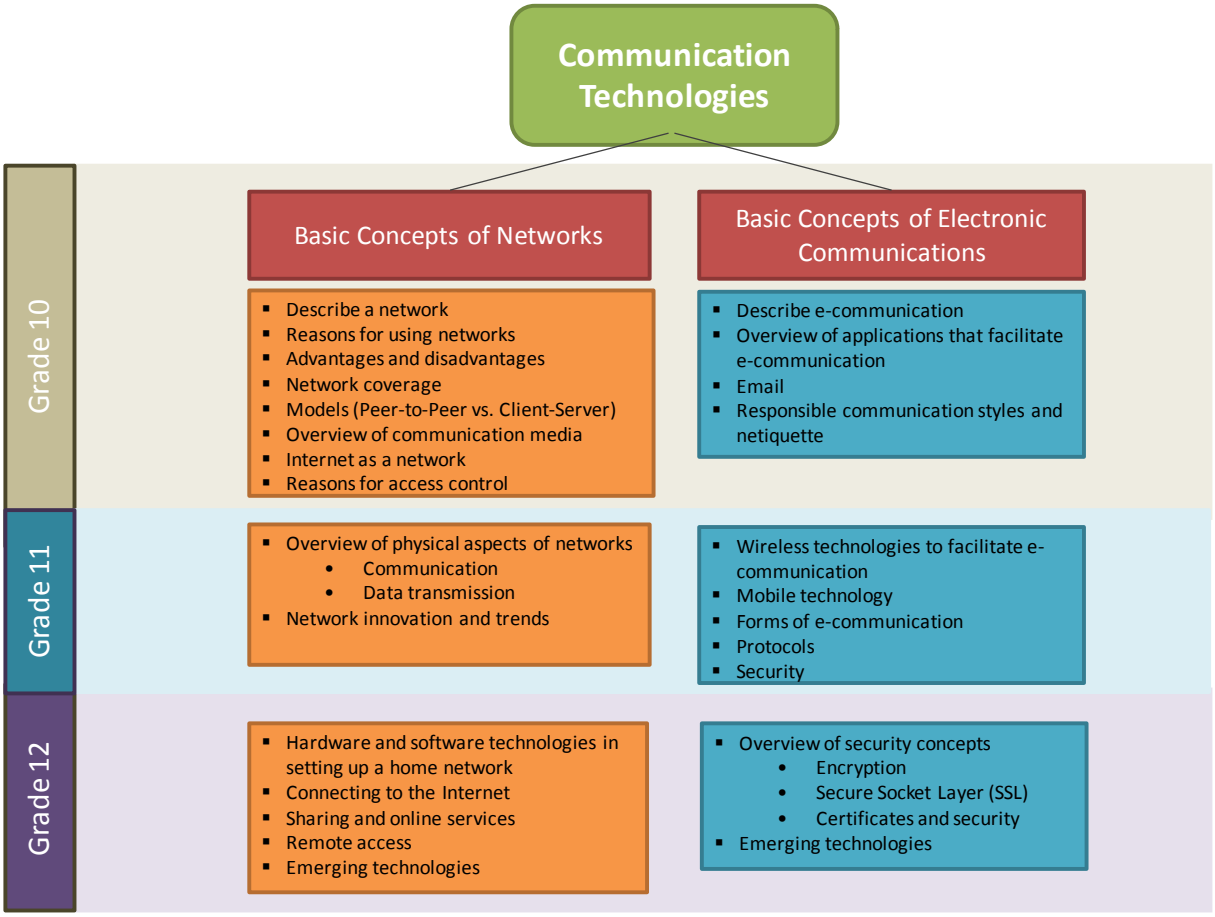
Learners should be able to use appropriate practices and tools to:

- solve computational problems through:
  - identifying and analysing requirements for a specific problem;
  - designing effective algorithms;
  - converting these to code; and
  - testing the solution to see if it meets the requirements.
- apply the principles of human computer interaction to design functional user interfaces.

3.2 Communication Technologies

Communication technologies include various network technologies to facilitate the management and dissemination of digital data from one point to another. Communication technologies also refer to the electronic systems used for electronic data interchange that facilitate, among others, communication and information dissemination between various individuals or groups at a single point or dispersed locations.

Broad topic layout and progression



Note:

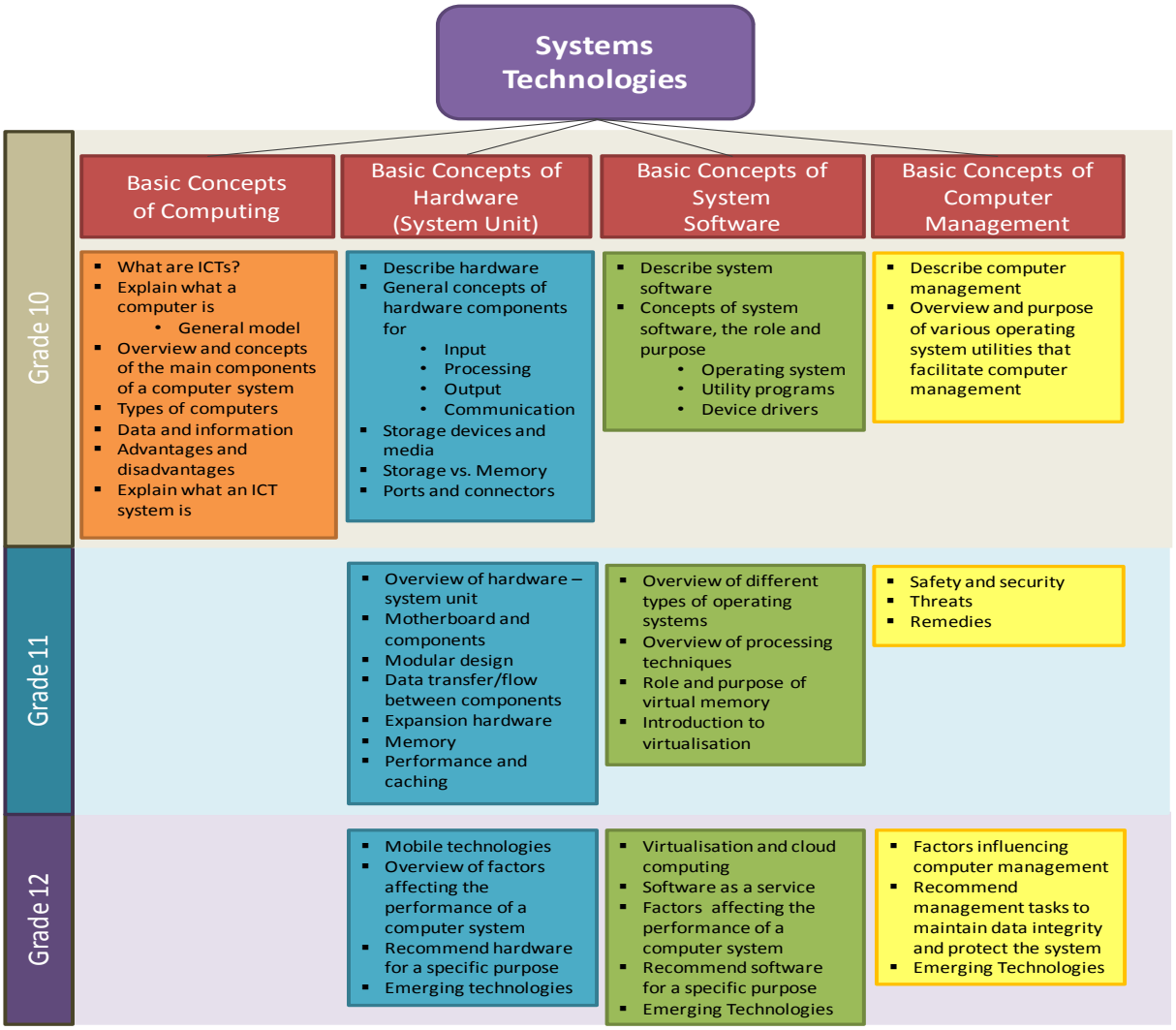
Communication Technologies should teach learners to:

- understand the concepts of the various technologies, standards and protocols involved in the electronic transmission of data via a computer-based network;
- understand the concepts of the technologies and standards implemented to enable electronic communication;
- understand the purpose and uses of communication software;
- understand how communications technology can benefit specific scenarios;
- be aware of and manage security issues; and
- be aware of new trends and developments.

3.3 Systems Technologies

Systems technologies refer to the physical and non-physical components of a computer system. The components of the system are generally related but unconnected in their original form. The connected components which include hardware, peripherals and software components allow the computer to perform the basic functions of a computing system. The basic functions of a computing system include input, processing, output, storage, communication and transfer of data in an electronic format.

Broad topic layout and progression



Note:

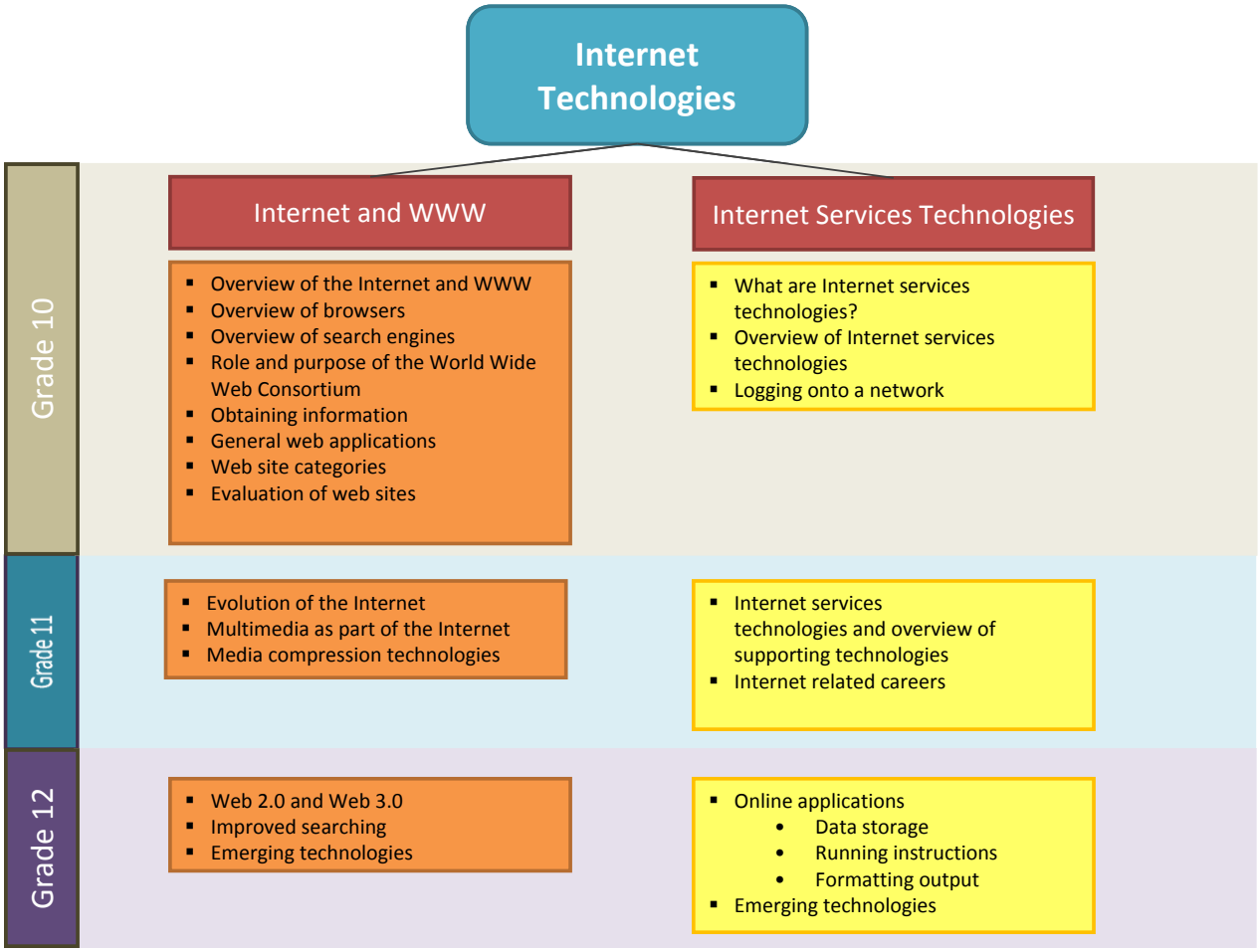
Systems Technologies should teach learners to:

- understand the hardware and software concepts that make up a computer system;
- make informed purchase decisions and whether to upgrade or buy new equipment or software;
- select the most appropriate hardware and software for a given scenario;
- understand how technology can benefit the user in specific contexts;
- understand the operations involved in the management and optimal utilisation of a computer system;
- troubleshoot at an elementary level; and
- be aware of new trends and developments and how to integrate these with existing or new equipment.

3.4 Internet Technologies

Internet Technologies are related and interconnected technologies which enable the establishment of global networks, for various purposes such as collaboration, electronic data interchange, electronic commerce and social networking. Internet services technologies refer to a range of technologies and tools for the design, development and maintenance of websites. The field of Internet services technologies includes Internet programming as well as the roles and responsibilities of each of the individuals involved. Internet technologies include the WWW and all interrelated processes in the digital presentation of multimedia data on a web page.

Broad topic layout and progression



Note:

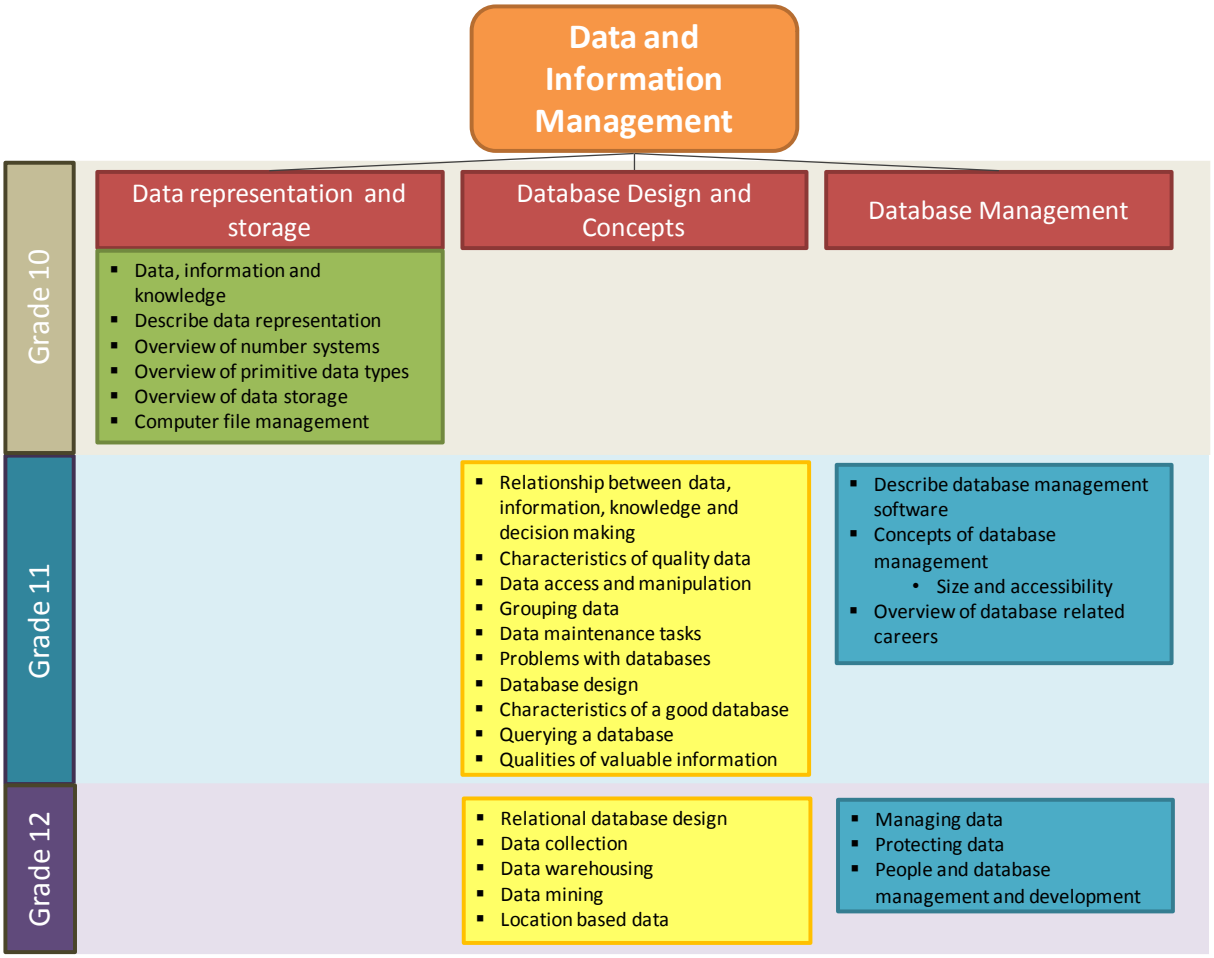
Internet Technologies should teach learners to:

- understand the role that the Internet and the WWW play as part of the global information super-highway and the contribution towards the digital age;
- understand the role of Internet services and supporting technologies;
- understand how Internet technology and services can benefit specific scenarios; and
- be aware of new trends and developments.

3.5 Data and Information Management

Data and information management refers to the techniques and technologies involved in the collection, storage, dissemination and processing of data into information that results in knowledge and leads to decision making. It includes database design principles with specific reference to data storage, retrieval and information presentation design.

Broad topic layout and progression



Note:

Learners need to develop an understanding of:

- data and information with regard to the representation and classification thereof;
- how business takes advantage of computer databases to store data and retrieve information that enables it to gain a competitive edge as well as the social, legal and ethical issues involved;
- database design for use as part of information-driven ICT systems and platforms; and
- DBMS software and its purpose and application in an information-driven society.

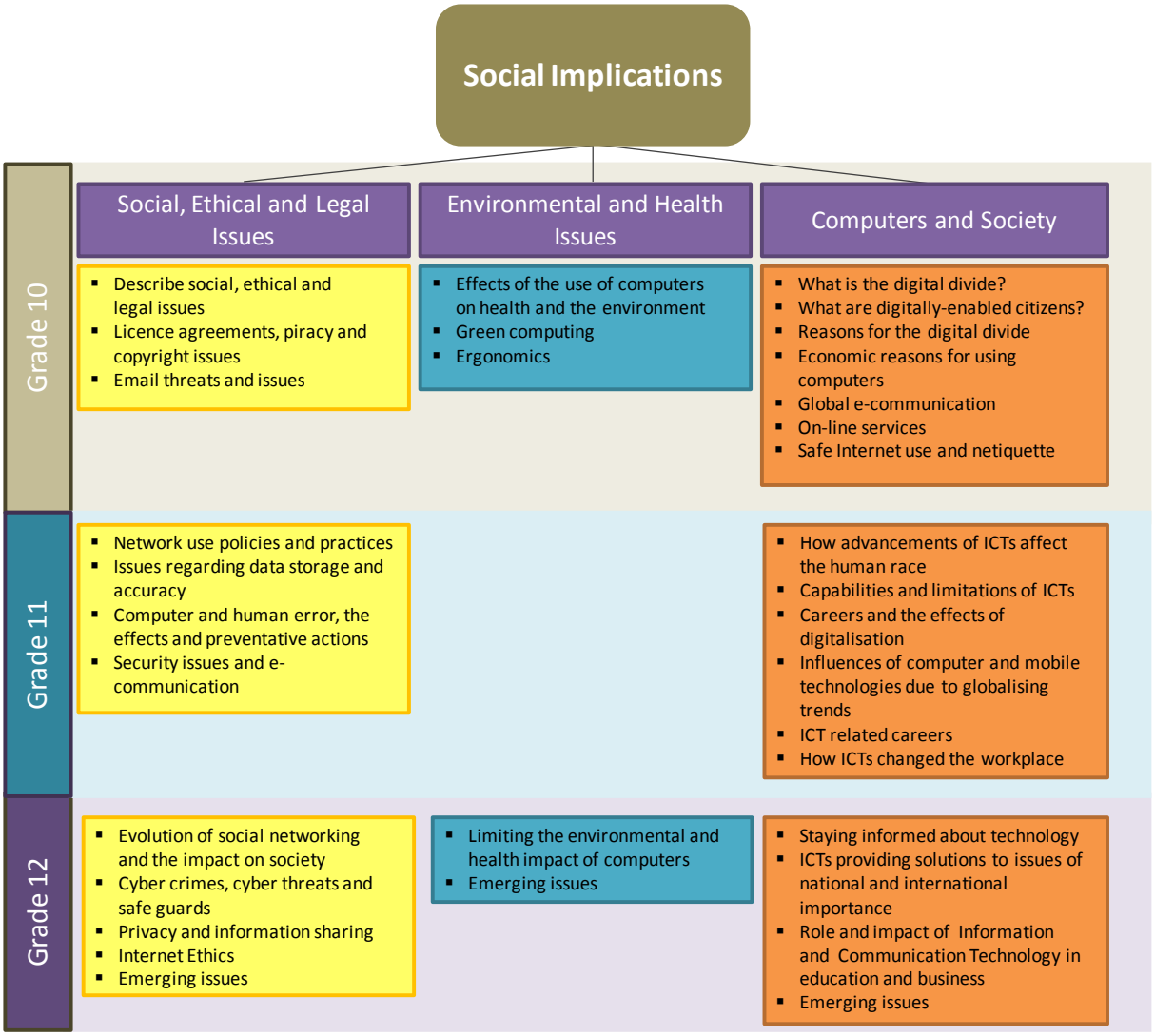
Database design, queries and reports should be linked to application development as described in the topic Solution Development.

This section also covers some practical aspects regarding learning about and working with databases.

3.6 Social Implications

Social implications in the IT curriculum refer to issues relating to the digital age, bridging the digital divide and the responsible use of ICTs.

Broad topic layout and progression



Note:

This topic should provide an overview and understanding of:

- social issues related to the use of computers and how ICTs affect modern life;
- risks and safety aspects that may be involved in the operation of computing equipment within a given context;
- risks and safety issues relevant to using the Internet; and
- principles for making informed decisions regarding the responsible use of ICTs.

Most of the content of Social Implications should be dealt with and integrated with other topics and should not be taught as a stand-alone topic. The time scheduled for this topic can therefore be added to other topics.

3.7 Suggested teaching plan

The suggested teaching plan indicates the minimum content to be covered per term. The sequence of the content or topics listed per term **is not prescribed.** Teachers should design their own work schedules (or use/adapt the work schedule provided in their textbook) to teach the content per term in **appropriate sequence** and pace.

The sub-topics presented in the term plans should not be seen as stand-alone topics. Relevant sub-topics or content should be presented in an integrated manner. Integrating the topics in the lesson presentation should flow naturally due to the nature, links and 'overlap' of the content. Some content from one sub-topic may strengthen and underpin the content of another. This approach should be applied throughout the three-year curriculum.

It is important that the specific technologies in the teaching plans are revised at regular intervals to phase out old technologies and to include new technologies.

As the length of terms varies from one year to another, the teaching plan/work schedules should be adapted accordingly on a year-to-year basis.

3.7.1 Grade 10

Grade 10: Term 1 – 10 weeks/40 hours

Systems Technologies: Basic concepts of computing (± 1 week/4 hours)

- What are Information and Communication Technologies (ICTs)?
- Define Information Technology
- Explain what a computer is: Overview of a general model of a computer in terms of input, storage, processing, output and communication
- Overview and concepts of the main components of a computer system:
  - Hardware vs software
  - Common/generic physical components of a home computer system: input (keyboard, mouse), storage (hard drive), processing (CPU and RAM), output (monitor, printer) and communication (modem/router)
  - Common/generic non-physical components of a home computer system: system software (operating system) and application software
    - Generic/common examples and uses
    - What are shareware, freeware, free open source software (FOSS), proprietary software?
  - Concept of interdependency of hardware and software
- Types of computers: desktop, notebooks, netbooks, tablets, smart phones, server, embedded computers (microcontrollers): purpose and uses
  - Differentiate between the types of computers in terms of primary uses, processing power and size
  - Categorise computers/classification of computers in terms of portability/mobility, processing power and usage
- Advantages and disadvantages of using computers
- Explanation of and differentiation between data and information:
  - Information processing cycle: input, processing, output, storage, communication (general concepts)
  - Transition from raw data to processed/organised information
  - Overview of uses and examples of information within an organisation
  - Why is information useful?
- What is an ICT system?
  - Overview of a general model of an ICT system: convey, manipulate and store data
  - Example of an ICT system (familiar context, e.g. point-of-sales system, cell phones)

Data and Information Management: Data representation and storage (±2 weeks/8 hours)

- Data, information and knowledge
- What is data representation?
- What is data storage?
- Bits and bytes
- Overview of number systems: decimal, binary, hexadecimal
  - Conversion between:
    - binary and decimal and vice versa
    - decimal and hexadecimal and vice versa
  - Overview of digital character representation, e.g. ASCII/UTF-8, Unicode
- Overview of primitive data types and their storage (integer types, text/string types, character types, floating point types)
- Overview of data structures and collections of data: storage in terms of:
  - Files, databases
  - Reasons for data storage
- Computer file management:
  - Organising files
  - Files, folders and drives
  - File specification: drive, path, filename, file extension
  - File manager
  - Hierarchical structure
  - Reasons for having a file structure
  - Manipulating files and folders
  - File-naming conventions
  - Common file types and extensions (association)
    - Archived and compressed
    - Forms of text files
    - Database, spreadsheet, presentations and word processing documents
    - Graphic files, movie, sound and animation files
    - Font files
    - Source code
    - Object code, executable files, shared and dynamically linked libraries

<table>
<tr><td colspan="2">Grade 10: Term 1 – 10 weeks/40 hours</td></tr>
<tr><td colspan="2">▪ Saving as another type/version and exporting between file types</td></tr>
<tr><td colspan="2">Social Implications (±½ week/2 hours)</td></tr>
<tr><td colspan="2">• Social issues applicable to term 1 content such as licence agreements (including creative commons), piracy, copyright, copyleft
• What are social, ethical and legal issues pertaining to ICTs?
• Economic reasons for using computers: saving paper, labour, communication costs, efficiency, accuracy and reliability
• Digital divide
▪ What is the digital divide?
▪ What are digitally enabled citizens?
▪ Reasons for the digital divide</td></tr>
<tr><td>Solution Development: Introduction to Algorithms (±2 weeks/8 hours)</td><td>Notes</td></tr>
<tr><td>• Basic concepts of an algorithm
▪ What is an algorithm? Develop a clear understanding
• Examples of algorithms in everyday life, e.g. instructions to draw a kite or fold a paper jet, recipe to bake a cake
▪ Devise an algorithm/basic instructions to complete similar tasks
▪ Use a tool, e.g. basic flowchart to describe a task
▪ Interpret a basic flow chart
• Explore algorithms such as:
▪ Determine smallest, largest value of more than two values
▪ Swapping values
▪ Determining aggregates, e.g. sum
▪ Basic calculations such as calculating area, volume, VAT
▪ Determine whether a number is even
▪ Determine whether a number is a factor of another number
• Produce an algorithm to solve a problem
• Tools, e.g. basic flow charts/pseudo code to represent an algorithm
• Trace an algorithm to determine the outcome– trace table
• Compare algorithms considering, e.g. order, precision and efficiency
• Value of accurate, well-tested algorithms</td><td>The purpose of this section is to serve as an introduction to solution development to develop the learner's computational thinking practices of algorithm development, problem solving and programming using everyday scenarios.
Exploring algorithms to solve generic problems will enable a learner to use similar principles to devise algorithms for new problems or situations. It will also enable the learner to identify the types of problems requiring certain generic algorithms.
Investigating specific algorithms should provide the learner with the opportunity to explore various ways to solve the same problem by using different principles or tools.</td></tr>
<tr><td>Solution Development: Introduction to solution development using a high level programming language (±4½ weeks/ 18 hours)</td><td>Notes</td></tr>
<tr><td>• Introduction to the programming tool – IDE/GUI, trace tables, basic terms and development environment
• Exploring the use of variables
• Variable naming conventions
• Assigning values to variables
• Exploring data types: integers, strings, floats, Boolean
• Casting (strToint, intTostr, floatTostr, strTofloat, strTofloatf)
• Operators (+ , , * , /) and order of precedence
• Retrieving remainders: modulus, div
• Functions – random, round, square root
• Basic calculations such as area, volume, VAT and simple formulae, typical calculations done in other subjects
• Formatting of output (fixed, currency)
• Basic string concatenation - + operator
• Applying algorithms such as swapping values, finding aggregates, isolate digits in an integer number
▪ Event handling (click)</td><td></td></tr>
<tr><td>Formal Assessment (PoA): Refer to Chapter 4 for mark and time allocation</td><td>Reporting</td></tr>
<tr><td>1 theory test covering content taught to that point</td><td>Theory test (100%)</td></tr>
</table>

Grade 10: Term 2 – 10 weeks/40 hours, including examination (2 weeks)

Systems Technologies: Basic concepts of hardware (± 1 week/4 hours)

- Describe hardware
- Extend hardware concepts
  - Input devices: purpose and uses (What is it? Where is it used? What is it used for?)
    - Alternative keyboards, pointing devices, touch-sensitive pads, pen input, video input, scanners, capturing devices, data collection devices, biometric input, card readers
  - Output devices: purpose and uses (What is it? Where is it used? What is it used for?)
    - Display devices, printers, data projectors
    - Concepts regarding quality of output and speed where applicable
  - Storage devices: purpose and uses (What is it? Where is it used? What is it used for?)
    - Hard drives (fixed and portable), USB flash drives, solid state drives (SSD), memory cards, optical disks (DVD and Blu-ray drives)
    - Capacity, portability, use
    - Technology used (magnetic, optic, electronic)
  - Input/Output devices: purpose and uses (What is it? Where is it used? What is it used for?)
    - Touch screens, game controllers, digital cameras, smart phone, smartboards, toy/electronic device interfaces
  - System unit (processing: Motherboard, CPU and RAM)
    - General function of CPU and RAM
    - Location of CPU and RAM
  - Identify ports and connectors and their purpose: USB, Thunderbolt, HDMI
  - Categorise hardware according to input, output, storage, processing
- Memory vs storage
- Compare input, processing, output, storage devices of a desktop computer with a small mobile device such as smart phone or tablet
- Which are the same? Which are different? Why are they the same/different?

Systems Technologies: Basic concepts of system software (± 1 week/4 hours)

- Describe system software
- Extend system software concepts
  - Operating system
    - What is an operating system?
    - What is the purpose/role of an operating system?
      - General role: group of related programs which manage hardware and software
      - Specific role: provides user interface, I/O management
      - Brief overview of the role of the operating system in terms of file, disk, memory, storage and process management
    - Types of operating systems (also associate with types of computers), e.g. stand-alone (home edition), network, embedded
    - Examples of common operating systems (Windows, Linux, iOS, Android)
  - Utility programs
    - What are utility programs?
    - What are they used for?
  - Purpose of device drivers

Communication Technologies: Networks (±½ week/2 hours)

- Describe a network
- Reasons for using networks such as communication, access to/sharing resources, centralisation, file and funds transfer, productivity, leisure
- Advantages and disadvantages of networks
- List the essential basic network components
- Overview of different communication media (wired vs wireless)
  - Types of cabling and components
  - Types of transmitters and components
- Local area network (LAN) vs. wide area network (WAN) – coverage and where it is used
- Internet as an example of a network (WAN)
- Differentiate between client-server and peer-to-peer networks
- Explain the reasons for logging into a network and connecting to a server – access control

Grade 10: Term 2 – 10 weeks/40 hours, including examination (2 weeks)

Communication Technologies: Electronic Communications (±½ week/2 hours)

- Describe electronic communication
- Overview of applications/tools to facilitate e-communication – purpose and uses (What is it? What is it used for?)
  - E-mail, web browser, File Transfer Protocol (FTP), instant messaging, chat rooms, video call and Voice over Internet Protocol (VoIP), Vlog, Blog, webinars
- E-mail as a form of e-communication
  - Uses of e-mail
  - E-mail accounts (Internet Service Provider (ISP) and web-based)
  - E-mail addresses
  - Use e-mail

Social Implications (±½ week/2 hours)

- Social issues applicable to term 2 content such as ergonomics, green computing issues, health issues
- Global e-communication, i.e. accuracy, time, distance, communication costs, speed

Solution Development: Software Engineering Principles (±1½ week/6 hours)	**Notes**

• What is problem solving?
• Problem solving steps *(Polya, G., 1957)*
  ▪ Understand the problem (task/problem description or scenario/user stories)
  ○ State in own words
  ○ Clarity on what needs to be done
  ○ What is known or given? What is missing or needed?
  ▪ Devise a plan/algorithm (textual)
  ○ Look for patterns
  ○ Look at related problems, known solutions
  ○ Examine simpler or special cases
  ○ Make a table, create diagram, use guess and check, work backwards, identify sub-goal
  ▪ Carry out the plan/implement the algorithm (write the code)
  ▪ Look back/test (see if it works)
  ○ Check results against original problem. Does it make sense? Is there another solution?
• Solve a problem using the problem solving steps
• Use appropriate tools and techniques used in software analysis, viz.:
  ▪ User stories (written by the client and provide the requirements)
  ▪ Noun-verb analysis of user stories
  ○ List of nouns provides identification of objects and state
  ○ List of verbs provides identification of behaviour
  ▪ Acceptance tests (does the program meet the requirements?)
	The purpose of this section is to teach problem-solving procedures and techniques.
	
	• Textual (algorithm) or flow chart (how to go about creating the animation/solving the problem)
	• Convert to programming code (write the program)
	• Test (see if it works)

Solution Development: Introduction to solution development using a high level programming language (±4 weeks/ 16 hours)	**Notes**

• Comparison operators and performing logical comparisons (from term 1)
• Conditional constructs (if and if-then-else) including Boolean operators
• Nested if's
• CASE statement
• Extend the use of variables, relational operators
• Boolean logic/operators (and, or, not)
• Strings
  ▪ String methods - length, setLength, UpperCase, UpCase
  ▪ String operations - comparing length, comparing strings
• Events – form create activate
• Basic validation techniques (input and processing), e.g. test for negative number when calculating square root
• Debugging techniques
• Debugging using the variable watch facility	

Formal Assessment (PoA): Refer to Chapter 4 for mark and time allocation	**Reporting**
1 practical test + 1 examination (1 practical paper + 1 theory paper)	Practical test (30%); Practical exam (35%), Theory exam (35%)

Grade 10: Term 3 – 10 weeks/40 hours

Systems Technologies: Computer Management (±1 week/4 hours)

- Describe computer management
- Overview and purpose of various management tasks and operating system utilities
  - Management of desktop
  - Management of files and folders
  - General housekeeping tasks
  - Defragmentation
  - Scheduling/updating
  - Archive, backup
  - Compress/decompress files
  - Security features – firewall, anti-virus, control of spyware, adware
  - Installing/uninstalling software (custom and full installation, product keys, activation codes)
  - Add devices/drivers – installation, Plug and Play
  - System settings and properties

Internet Technologies: Internet and WWW (±1½ week /± 6 hours)

- Overview of the Internet
- Describe the Internet
- Internet addresses – Internet protocol (IP) addresses and domain names
- What is needed to connect to the Internet referring to
- Internet Service Providers (ISPs), wired and wireless connections
- Overview of the World Wide Web (WWW)
- Describe the WWW
- Web address/uniform resource locater (URL)
- Web page and website
- Types of websites, their purpose/what they offer and examples
  - Weblog/Vlog (blog), Wiki, social network, web applications (e.g. Google docs, OneDrive, Google drive, Office 365)
- Criteria to evaluate websites
- Affiliation (e.g. who supports the website?)
- Audience (e.g. level at which it is written/who is it intended for?)
- Authority (e.g. who is the author and what are his/her credentials?)
- Content (e.g. organisation of content and working links)
- Currency (e.g. is the information on the web page up-to-date?)
- Design (e.g. is it easy to navigate and visually pleasing? How quickly does it download?)
- Objectivity (e.g. does it reflect any preconceptions?)
- Browsing and searching
- Examples of web browsers
- What is a search engine?
- Examples of search engines
- Performing searches using a search engine (search techniques)
- How to access and browse a website
- What is the World Wide Web consortium (W3C)?

Social Implications (±½ week/2 hours)

- Responsible communication styles and netiquette
- Social issues applicable to term 3 content such as:
  - E-mail threats and issues: viruses, hoaxes, spam, phising, e-mail spoofing and pharming
  - Safe email and Internet use: dangers and tips to ensure safe use

Solution Development: Software Engineering Principles and Practical Assessment Task (PAT) (±2 weeks/8 hours)	**Notes**
Start with PAT and reinforce software engineering principles, problem-solving techniques and algorithms as well as debugging techniques (see PAT guidelines) • Scenario/problem statement, research and requirements • Visual [state-transition-diagram], textual [algorithm] or flow chart (how to go about creating the GUI/solving the problem)	

Grade 10: Term 3 – 10 weeks/40 hours	
Solution Development: Introduction to solution development using high level programming language (±5 weeks/ 20 hours)	**Notes**
Using good programming principles and algorithmic development extend the use of the tool: • Iteration constructs (for), pre-conditional and post-conditional (While, repeat until and for-loop **Conditional iteration** • Extend iteration ▪ Sentinel-controlled loops – flag (from grade 11) • String handling – position, copy, delete, insert from first principles (no built-in methods) • Develop an elementary game or other suitable programs that exercise the content of the syllabus • Develop simple applications incorporating a combination of graphics, iteration, conditional constructs, concepts covered • **Input and output using a text file** • Apply simple file input and output using a text file to populate data structures and to develop simple reports ▪ Text files are incorporated utilising text stream operations and methods which load and save a file stream, etc. ▪ Accessing text stream operations and methods to load and save a file stream, etc. ▪ Utilise exceptions to catch errors on input and output	Implement basic algorithms to solve general computing problems using methods such as: • Determine whether a number is a prime number • Lowest common multiple (LCM), greatest common divisor (GCD) • Find a specified character in a string • Find/extract a substring/ character in a string • Count the number of occurrences of a specific character in a string Exploring algorithms such as to convert binary numbers, digital character representation Develop and use algorithms to solve various problems
Formal Assessment (PoA): Refer to Chapter 4 for mark and time allocation	**Reporting**
1 practical test + 1 alternative task	Practical test (50%); Alternative task (50%)

<table>
<tr><td colspan="2">Grade 10: Term 4 – 10 weeks/40 hours, including examination (3 weeks)</td></tr>
<tr><td colspan="2">Internet Technologies: Internet and WWW (±½ week/2 hours)
• Overview of plug-in applications
▪ Describe plug-in applications
▪ Examples and purpose of plug-in applications for browsers such as PDF converters and tools, Flash player, Java, QuickTime player, RealPlayer, Silverlight
• What are Internet services technologies?
• Usability of web pages/sites
▪ Compare usability issues such as readability, navigation, consistency, layout, typography
▪ How does this relate to user interface design?
▪ Concept of a web page as a file that contains text and HTML and/or XHTML code</td></tr>
<tr><td>Solution Development: Introduction to solution development (±1 weeks/ 4 hours)</td><td>Notes</td></tr>
<tr><td>Using good programming principles and algorithmic development extend the use of the tool:
• Revise, consolidate and extend solution development content by developing applications incorporating a combination of features
Text-based reports
• Generating a simple text-based report, e.g. summary of data</td><td></td></tr>
<tr><td>Solution Development: Software Engineering Principles and Practical Assessment Task (PAT) (±3 weeks/12 hours)</td><td></td></tr>
<tr><td>• Finalise PAT
▪ Construct a solution based on the planning
▪ Document the solution by adding comments</td><td></td></tr>
<tr><td>Formal Assessment (PoA): Refer to Chapter 4 for mark and time allocation</td><td>Reporting (promotion mark)</td></tr>
<tr><td>1 examination (practical paper + theory paper) + PAT</td><td>Convert: PAT to 25%
Paper 1 to 25%
Paper 2 to 25%
Term 1 + Term 2 + Term 3 marks to 25%</td></tr>
</table>

Grade 11

Grade 11: Term 1 – 10 weeks/40 hours

Systems Technologies: Hardware (±1 week/4 hours)

Extend hardware concepts from Grade 10:

- Describe the motherboard
- Purpose and role of the motherboard
- Components as part of the motherboard
  - Purpose and role of a BIOS chip, CPU, GPU, RAM, VRAM, ROM, slots, cards and buses
- Modular design
- Flow/transfer of data between components
  - Busses (USB)
  - Point-to-point connections
  - Purpose and role of cache memory and caching
  - Storage → RAM → CPU
  - RAM → VRAM → GPU
- Purpose and role of the expansion cards
- Memory as part of a computer system
  - ROM, RAM – role and characteristics
  - Temporary/permanent/magnetic/optic/solid state
- Difference in performance of different components and caching (cache memory, web caching and disk caching)

Systems Technologies: Software (±1 week/4 hours)

Extend functions of system software from Grade 10:

- Various types of operating systems in terms of cost, size, hardware needed and platform
  - Programming language compilers/interpreters
    - What are programming language compilers/interpreters?
- Overview of processing techniques (managed by systems software)
  - Multi-tasking, multi-threading, multi-processing (Definition, comparison)
- What is virtual memory? Role and purpose
- Introduction to virtualisation – overview
  - Describe virtualisation
  - Virtual machines – purpose

Communication Technologies: Networks (±1 week/4 hours)

- Overview of physical aspects of a network
  - Communication (Wi-Fi, WiMAX, 3G, LTE)
  - Data transmission
    - Media (reinforce from Grade 10)
    - Physical layout (topology – star)
    - Physical limitations (bandwidth)
    - Connection (NIC, modem, switch, router/bridge)
    - Size (PAN/HAN, LAN, WAN)
- Overview of network innovation
  - VoIP
  - Virtual Private Networks (VPN)
  - Location-based computing (3G, LTE, wireless, GPS)

Social implications (±½ week/2 hours)

- Social issues applicable to term 1 content such as network use policies and practices
- How the advancement of ICT affects the human race
  - Computers providing solutions to issues of national and international importance such as
    - Weather, elections, census
- Capabilities and limitations of ICTs

Grade 11: Term 1 – 10 weeks/40 hours

Systems Technologies: Computer Management (±½ week/2 hours)

Extend computer management issues regarding safeguarding against threats

- Safety and security
  - Human error (GIGO, accidents)
- Threats
  - Physical access
    - Theft
    - Flash drives and portable media
  - Hardware failure
    - Storage
    - Power
  - Network vulnerability
    - Virus, worm, Trojan, rootkit, spoofing, phishing
- Remedies
  - Backup (including on-line storage), UPS, passwords, access rights, firewalls, anti-virus, validation

Solution Development: Application Development using a high-level programming language (±6 weeks/24 hrs)	**Notes**
• Explore nested loops (from grade 10) **Arrays** • Arrays as a data structure (1-dim) • Structure • Step through items • Basic operations, e.g. aggregates • Searching using the linear search algorithm • Sorting an array (a sorting algorithm) • Parallel arrays **String and Date manipulation** • String manipulation using string methods: • inserting and deleting characters • determine the position of a character • find a character/substring • determine the length of a string • Extend applications by adding built-in methods to perform simple string manipulation • Create date and time objects • Changing the date and time • Formatting date and time • Date calculations • Date methods (TimeToStr, DateToStr, IsLeapYear) **Built-in methods** • Extend the use of built-in methods and the concept of parameters/message passing	
Formal Assessment (PoA): Refer to Chapter 4 for mark and time allocation	**Reporting**
1 practical test + 1 theory test	Practical test (50%); Theory test (50%)

Grade 11: Term 2 – 10 weeks/40 hours, including examinations (2 weeks)

Communication Technologies: Electronic Communications (±1 week/4 hours)

- Mobile/wireless e-communication
  - E-mail and blogging
  - Micro blog, SMS, instant messaging
  - Media: video casting, podcasting, VoIP, video conferencing
- Use of mobile technology
  - Mobile devices such as cell phones, smart phones, future phones
  - Mobile browser – description
- Use of wireless technologies
  - Access points
  - GPS, 3G, 4G, WiMAX, Bluetooth, etc.
  - Difference in range and bandwidth (non-technical)
- Protocols
  - How protocols control data (POP3, SMTP, VoIP)
- Security
  - Passwords, firewalls, encryption

Social Implications (±½ week/2 hours)

- Social issues applicable to term 2 content such as social engineering, impact of social websites
- List and discuss computer and human error and the effects thereof such as:
  - Accuracy and validity – data input
  - Data types used
  - Verification and validation of data
  - Software bugs
  - Hardware failure
  - Hardware configurations
- Preventative actions

Data and Information Management: Database Management (±½ week/2 hours)

- Describe database management software (DBMS)
- Examples of DBMS software, e.g.
  - Microsoft SQL Server
  - Oracle
  - Microsoft Access
  - Blackfish
  - Open source databases (PostgreSQL, MySQL)
- Database management – size and accessibility
  - Desktop vs server (size and accessibility)
  - Distributed (e.g. Google)
  - DBMS software
- Overview of database-related careers and roles of people involved
  - DBA (database administrator)
  - Database Programmers
  - Database Analysts
  - Database Project managers

Solution Development: Software Engineering Principles and Problem solving (±1 week/4 hours)	**Notes**
• What is software development? • Planning and implementing a solution ▪ Define/understand the problem/task o Read the specifications and analyse the problem/task to determine the requirements ▪ Design the interface and the solution o Develop a logical solution based on the specifications and analysis as well as sound software engineering principles o Consider functionality and usability issues in designing the interface ▪ Code/implement o Incorporate suitable programming constructs in the development of a solution ▪ Test and debug the program	

Grade 11: Term 2 – 10 weeks/40 hours, including examinations (2 weeks)	
○ Use testing and debugging techniques and methods ▪ Document, implement and maintain the program • Planning techniques using any appropriate tools	
Solution Development: Application Development using a high-level programming language (±5 weeks/20 hours)	**Notes**
Input and output using a text file ▪ Extend the use of textfiles on delimited strings **Extended Text-based reports** • Generating a text-based report, e.g. correctly formatted data Using good principles of algorithmic development and problem solving, extend programming with high-level language: **User defined methods with and without parameter passing** ▪ Procedures – without reference parameters ▪ Functions ▪ Arguments vs parameters ▪ Dynamic Instantiation of active and passive components	
Formal Assessment (PoA): Refer to Chapter 4 for mark and time allocation	**Reporting**
1 examination (1 practical paper + 1 theory paper)	Practical exam (50%); Theory exam (50%)

Grade 11: Term 3 – 10 weeks/40 hours

Data and Information Management: Database design concepts (±3 weeks/12 hours)

- Relationship between data, information, knowledge and decision making
- Characteristics of quality data:
  - Accuracy, correctness, currency, completeness, relevance
  - Data validation, e.g. format check, data type check, range check, check digit
- Qualities of valuable information
- How to get to information
  - Accessing and manipulating data
    - Manual
    - Electronic
- Grouping data
  - Records and fields
  - Different types of fields and their purpose, e.g. primary key, alternate key
  - Tables
  - Relationships
- Create a simple database with focus on table design without relationships
- Data maintenance tasks such as
  - Insert/add, delete, edit
  - Process, sort, query (generating information from a database)
- Set up relationships between tables
  - 1:M e.g. register class → pupils
  - Two tables showing master detail relationship with at least one foreign key in one table
  - Primary key and foreign key
- Query a database using a join on a maximum of two tables with multiple criteria (the database may contain more than two tables, however a maximum of two tables is joined for query purposes)
- Simple entity relations diagrams (ERD

Social Implications (±½ week/2 hours)

- Social issues applicable to term 3 content
- IT-related careers and the effects of digitalisation
  - Careers: PC technician, programmer, network administrator, graphics design, web authoring, security consultants, systems analyst
  - Effect on workplace and employment practices
  - Mobile offices, virtual office, decentralisation of labour, office automation, robotics, artificial intelligence
  - Ability to balance the advantages and disadvantages of a computerised system

Solution Development: Software Engineering Principles and PAT (±2 weeks/8 hours)

- Start with Practical Assessment Task
  - Reinforce problem-solving steps
  - Reinforce software engineering principles

Grade 11: Term 3 – 10 weeks/40 hours

Solution Development: Application Development using a high-level programming language (±4½ weeks/18 hours)

Using good principles of algorithmic development and problem solving, extend programming to incorporate database programming:

- Accessing a database through programming language constructs
- Set up a connection or connect to a database (single table) by providing path in code statements
- Develop a multi-form/multi-screen GUI incorporating simple controls – consider functionality and usability
- Use programming language constructs in the execution of various simple database transactions
  - Access fields and records within a dataset with code constructs and applicable methods
  - Navigate the records of a dataset
  - Modify individual fields and records within a dataset with code constructs and applicable methods, and apply all changes
  - Manipulate a dataset object and records with code constructs and apply all changes
  - Incorporate dataset event handlers and methods as part of the solution (only used in PAT)
  - Reinforce concepts such as iteration and conditions
- Use common dataset event handlers in the development of a solution (only used in PAT)
- Reinforce methods as part of a solution
- Apply simple parameter passing and return values using class methods as part of the form class
- Design and develop solutions for specific problems that include computational thinking and applying software engineering principles
  - Apply generic algorithms as part of the solution
  - Incorporating database transactions managed by methods or events
  - Devise a specific algorithm where applicable to solve a problem utilising user-defined code constructs or built-in methods
  - Motivate the use of a specific algorithm
  - Validate the solution against a set of data using different techniques, e.g. trace tables, watches, manual output comparison

Design and develop solutions for specific problems that include computational thinking and applying software engineering principles using event-driven programming within the OOP paradigm which may include database connectivity as part of the solution

Formal Assessment (PoA): Refer to Chapter 4 for mark and time allocation	**Reporting**
1 practical test + 1 alternative task	Practical test (50%); Alternative task (50%)

Grade 11: Term 4 – 10 weeks/40 hours, including examinations (2 weeks)

Internet Technologies: Internet and WWW (± ½ week/2 hours)

- Overview of the evolution of the Internet in terms of:
  - Software and applications
    - WEB 1.0, WEB 2.0, WEB 3.0
  - Technology
    - Fixed location vs mobile
- Overview of multimedia as part of Internet technologies
  - Download vs streaming
  - Live broadcasts
  - Video on-demand and IPTV (Internet Protocol Television)
- Media
  - Compression technology (MP3, Mpeg4, Mpeg2, Jpeg)
  - Compression: Quality vs bandwidth and speed

Internet Technologies: Internet Services Technologies (±1 week/4 hours)

- Overview of Internet services technologies
  - Types of websites (i.e. what they offer)
    - Static vs dynamic sites (ability to store data, interactivity, media, advantages and disadvantages)
    - Location based services sites
    - Internet sites' accessibility to mobile devices
- Overview of supporting technologies:
  - HTTP, HTTPS, VoIP, SEO (search engine optimisation)
  - Rich Internet applications
  - Security services
- Internet vs Intranet vs Extranet
- Internet related careers
  - Web designer
  - Web author
  - Graphics and multimedia designer

Social Implications (±½ week/2 hours)

- Social issues applicable to term 4 content
- Describe the influences of computer and mobile technologies on society due to globalising trends
  - Online services (online banking, booking reservations, e-learning)
  - Video conferencing, interactive whiteboards, online banking, cell phone banking, social websites (e.g Facebook)

Solution Development: Software Engineering Principles and PAT (±3 weeks/12 hours)

- Develop solutions for various problems using computational thinking and applying software engineering principles that include both database and non-database problems
  - Test and validate a solution against a set of design specifications
  - Alter a solution to meet a set of design specifications
  - Document a solution design and development
  - Motivate the design and development of the solution

Evaluate a solution against other solutions

- Use algorithmic thinking and software engineering principles to develop solutions for a variety of problems, focusing on computational problems which could include a database as part of the solution:
  - Apply generic algorithms as part of the solution
  - Devise a specific algorithm where applicable to solve a problem utilising user-defined code constructs or built-in methods
  - Contrast generic algorithms to built-in methods

Validate the solution against a set of data using different techniques, e.g. trace tables, watches, manual output comparison

- Reinforce software engineering principles, algorithms and problem-solving techniques
- Practical Assessment Task – finalise

Formal Assessment (PoA): Refer to Chapter 4 for mark and time allocation	**Reporting**
1 examination (1 practical paper + 1 theory paper) **Practical Assessment Task (PAT)**	Convert: PAT to 25% Paper 1 to 25% Paper 2 to 25% Term 1 + Term 2 + Term 3 marks to 25%

Grade 12

Grade 12: Term 1 – 10 weeks/40 hours

Data and Information Management: Database Management (±1 week/4 hours)

- Caring for and managing data
  - Value of data
  - How to protect data: validation, verification, integrity, logging changes (audit trail) – (who, what, when) warehousing, controlling access (passwords, security, user rights), parallel data sets
- Hacking through data
  - Invalid/false data
  - Database management software (DBMS) flaws (SQL injection)
- Differentiate and list the roles of people as part of database management, and database systems development
  - Database administrator (DBA)
  - Programmer
- Data collection – Overview and examples
  - RFID
  - Online
  - Invisible (e.g. credit card, loyalty card, government, forms, toll road passes, cellphone)
- Data warehousing
  - Describe data warehousing
  - Purpose and uses
- Data mining – description and purpose
  - SQL
  - Extracting data
  - Looking for patterns
  - Discovering knowledge
  - Strategy
- Location-based data

Data and Information Management: Database design concepts (±2½ weeks/10 hours)

- What is transaction processing system with regard to various database transactions?
- Characteristics of a good database
  - Data integrity
  - Data independence
  - Data redundancy
  - Data security
  - Data maintenance (ease of)
- Problems with databases
  - Anomalies
  - How to get rid of anomalies (concept of normalisation)
  - Split tables and create relations
  - Key fields
    - Reinforce primary and alternate keys
    - Foreign keys
    - Composite keys
  - Example of basic relationship enabled by the utilisation of key fields
- Design guidelines
- Design and create a relational database
- Explain and motivate relational database design
  - Relational database overview
    - Normalisation (overview and purpose) to reduce data redundancy and limit data anomalies
    - Where does un-normalised data come from?
      - Analyse general documents, e.g. a till slip to identify possible data entities
- Design/entities, keys, record organisation

Systems Technologies: Hardware (±½ week/2 hours)

- Mobile technologies
  - Examples: Smart phones, laptops, tablets, phablets
  - Advantages of mobility (size, weight)
  - Constraints (battery life, computing power vs. power consumption)
- Overview of factors influencing performance of a computer
  - CPU (speed and multi processing)
  - Memory capacity (cache and RAM)

<table>
<tr><td colspan="2">Grade 12: Term 1 – 10 weeks/40 hours</td></tr>
<tr><td colspan="2">

Storage speed
Network speed

Motivate a typical computer system in respect of the hardware needed for a specific purpose

Computer system for

Home/personal use
Game and entertainment
SOHO (Small-Office-Home-Office) user
Power user

</td></tr>
<tr><td colspan="2">Social Implications (±½ week/2 hours)</td></tr>
<tr><td colspan="2">

Social issues applicable to term 1 content, e.g. reducing the environmental impact of the use of computers could be reduced
Discuss various ways to stay informed about computer technology
Getting latest product upgrades, viruses and other threads, upgrading

</td></tr>
<tr><td>Solution Development: Application Development using high-level programming language (±4 weeks/16 hours)</td><td>Notes</td></tr>
<tr><td>
Extend database and programming to incorporate relational databases

Accessing a relational database through a programming language
Query a database using a join on a maximum of two tables with multiple criteria (the database may contain more than two tables, however a maximum of two tables is joined for query purposes) – reinforce:

Reinforce concepts, programming skills, algorithms and problem-solving skills developed in Grades 10 and 11 by means of application development

Develop a simple user-defined class to meet the program specifications as part of the solution
Instantiate a user-defined object as part of the solution
Reinforce method invocation
Differentiate between various types of methods in relation with their use and purpose (constructors, destructors, accessors, mutators, auxiliary)

Extend database and programming

Design and develop a solution incorporating SQL

Select, distinct
Insert, update, delete
Where
Order by
Group by
Special operators: Between, In, Like, Is Null, Having
Creating calculated fields, concatenating fields
Formatting with round, int, etc.
Casting a field
Create a join query (single joins) using ‘Where’
Mathematical operators
Aggregate functions: Sum, Average, Min, Max, Count
Common date functions
String functions (Length, Mid, Left, Right)

</td><td></td></tr>
<tr><td>Solution Development: Software Engineering Principles and PAT (±1½ weeks/6 hours)</td><td>Notes</td></tr>
<tr><td>

Start with PAT – Task description and analysis of requirements using an appropriate methodology

</td><td></td></tr>
<tr><td>Formal Assessment (PoA): Refer to Chapter 4 for mark and time allocation</td><td>Reporting</td></tr>
<tr><td>1 alternative task + 1 theory test</td><td>Alternative task (50%);
Theory test (50%)</td></tr>
</table>

Grade 12: Term 2 – 10 weeks/40 hours, including examinations (2 weeks)

Systems Technologies: Computer Management (±½ week/2 hours)

- Factors influencing computer management
- Recommend management tasks for general housekeeping and to maintain data integrity and protect the system

Systems Technology: Software (±1 week/4 hours)

- Overview of cloud computing and virtualisation
  - Describe cloud computing
  - Effect on hardware needs
  - Software as a service (SaaS)
    - Description and advantages
    - Who owns what?
  - Virtualisation of devices
    - What is it
    - What is it used for
  - Arguments for and against

Social Implications (±½ week/2 hours)

- Computer criminals
  - Hackers, crackers, cyber gangs, virus authors
- Types of cyber crimes
- Effect of cyber crimes
- Computer crimes such as hardware, software, information, identity theft, bandwidth theft, theft of time and services
  - Internet-related fraud scams
  - Internet attacks (worms, virus, denial of service, back doors)
  - Phishing
  - Unauthorised remote control and administration, e.g. botnets, zombies
  - Right to access vs right to privacy, misuse of personal information
- Safeguards against computer crimes, threats and criminals

Solution Development: Application Development using a high-level programming language (±4 weeks/16 hours)

Developed in Grades 10 and 11 by means of application development

- Develop a simple user-defined class to meet the program specifications as part of the solution
- Instantiate a user-defined object as part of the solution
- Reinforce method invocation
- Differentiate between various types of methods in relation with their use and purpose (constructors, destructors, accessors, mutators, auxiliary)
- Arrays as a data structure (2-dim)
  - Structure
  - Step through items
  - Basic operations, e.g. row/column aggregates

Use algorithmic thinking and software engineering principles to develop solutions for a variety of problems, that include both database and non-database problems

Solution Development: Software Engineering Principles and PAT (±2 weeks/8 hours)

- Reinforce software engineering principles
- Interface design: Functionality and usability principles and program design
- Practical Assessment Task – continue

Formal Assessment (PoA): Refer to Chapter 4 for mark and time allocation	**Reporting**
1 practical test + examination (1 practical paper + 1 theory paper)	Practical test (30%); Practical exam (35%); Theory exam (35%)

Grade 12: Term 3 – 10 weeks/40 hours, including examinations (3 weeks)

Internet Technologies: Internet Services Technologies (±½ week/2 hours)

- Improve searching
  - Semantic search
  - Mediated search
- Online applications
  - Storing data
    - Locally (cookies)
    - Online (databases)
    - Role of SQL, scripting languages (e.g. PHP, JavaScript), XML
  - Running instructions
    - Locally (scripts, AJAX)
    - Online (server side, scripts and code)
  - Formatting output
    - CSS

Communication Technologies Networks (±½ week/2 hours)

- Setting up a network
  - Essential parts
    - Switch, cables, wireless base station
  - Connecting to the Internet
    - Router/modem, ADSL/Wimax/3G
  - All-in-one solution ('router' is modem, router, switch and base station – all in one)
- Sharing concepts
  - Sharing files and folders, user rights, BitTorrent (Risks and benefits)
  - Online services (Drop box/Mobile Me/GSuite/Office 365)
- Remote access
  - On local network, through Internet, VPN

Communication Technologies: E-communications (±½ week/2 hours)

- Overview of security concepts
  - Encryption
  - SSL (private and public key)
  - Certificates and security

Social Implications (±½ week/2 hours)

- Social issues applicable to term 3 content
- Explain how computers provide solutions to issues of national and international importance such as:
  - Distributed computing power
  - Decision making
- Describe the evolution of social networking and the effect on society:
  - Information overload
  - Availability of personal information
    - Consequences of search engines and group communications
    - Social, political, environmental
    - Global community – cultural effects
    - Social websites and social engineering
    - Wikis
- List and discuss issues regarding privacy and information sharing
  - Cookies, anonymity, Global Unique Identifiers, file sharing – movies, music

Solution Development: Application Development using a high-level programming language (±2 weeks/8 hours)

- Consolidate and reinforce content, concepts and skills
- Design and develop solutions for a variety of problems that include computational thinking and applying software engineering principles
  - Test and validate a solution against a set of design specifications
  - Alter a solution to meet a set of design specifications
  - Document a solution design and development
  - Motivate the design and development of the solution
  - Evaluate a solution against other solutions

Solution Development: Software Engineering Principles and PAT (±3 weeks/12 hours)

- Reinforce software engineering principles
- Practical Assessment Task – Finalise

Formal Assessment (PoA): Refer to Chapter 4 for mark and time	**Reporting**

Grade 12: Term 3 – 10 weeks/40 hours, including examinations (3 weeks)	
allocation	
Examination (1 practical paper + 1 theory paper)	Practical exam (50%) Theory exam (50%)

Grade 12: Term 4 – 10 weeks/40 hours, including examination (7 weeks/28 hours)	
Content using Case Studies – All topics (±1½ weeks/6 hours)	
Consolidate content, concepts and skills using case studies to: • Identify the basic hardware configuration of a computer in terms of the processor, memory and hard drive size • Understand computers and their uses • Know how to use computers as tools to access information and to communicate with others around the world • Make better buying decisions – interpret advertisements and make judgements about quality and usefulness when buying equipment and software • Know how to fix simple computer problems and deal with challenges that arise with utilising computers (and know when to call for help) • Know what kind of computer uses could benefit and advance work place and career path opportunities • Know how to protect themselves against online villains and threats • Know how to apply digital tools to communicate, gather, analyse, use information and solve problems • Understand technology concepts, systems and operations • Recommend specific hardware/software for a specific scenario	
Solution Development: Application Development (±1½ weeks/6 hours)	
Consolidate content, concepts and skills to develop a software solution	
External examination (±7 weeks/24 hours)	
▪ Practical examination	25%
▪ Theory examination	25%
External examination: 1 practical paper + 1 theory paper Practical Assessment Task (PAT)	

Section 4

Assessment in Information Technology

4.1 Introduction

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 in order to improve the process of learning and teaching.

Assessment involves activities that are undertaken throughout the year. In grades 10 – 12 assessment comprises two different but related activities: informal daily assessment (assessment for learning) and formal assessment (assessment of learning).

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

4.2 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' progress. This is done through observation, discussion, practical demonstrations, learner-teacher conferences, informal classroom interactions, etc. Informal assessment may be as simple as stopping during the lesson to observe learners or to discuss with learners how learning is progressing. Informal assessment should be used to provide feedback to the learners and to inform planning for teaching, but need not be recorded. It should not be seen as separate from learning activities taking place in the classroom. Learners or teachers can mark these assessment 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.

4.3 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.

The following tables provide the formal assessment requirements for Information Technology:

Grade 10

Formal Assessment			
During the Year	**End-of-Year Examination**		
25%	**75%**		
SBA tasks	**Practical Assessment Task**	**End-of-Year Exam Papers (50%)**	
25%	**25%**	**25%**	**25%**
• 3 tests • 1 task • 1 exam (mid-year)	Project Software development project including aspects of planning cycle as well as principles of software engineering	Written exam 2.5 hours Theory aspects of all content, concepts and skills of all topics	Practical exam 3 hours Solution Development

Grade 11

Formal Assessment			
During the Year	**End-of-Year Examination**		
25%	**75%**		
SBA tasks	**Practical Assessment Task**	**End-of-Year Exam Papers (50%)**	
25%	**25%**	**25%**	**25%**
• 3 tests • 1 task • 1 exam (mid-year)	Project Software development project including aspects of planning cycle as well as principles of software engineering	Written exam 3 hours Theory aspects of all content, concepts and skills of all topics	Practical exam 3 hours Solution Development

Grade 12

Formal Assessment			
During the Year	**End-of-Year Examination**		
25%	**75%**		
SBA	**Practical Assessment Task**	**End-of-Year Exam Papers (50%)**	
25%	**25%**	**25%**	**25%**
• 2 tests • 1 task • 2 exams (mid-year and trial)	Project Software development project including aspects of planning cycle as well as principles of software engineering	Written exam 3 hours Theory aspects of all content, concepts and skills of all topics	Practical exam 3 hours Solution Development

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.

4.3.1 Types of formal assessment for Information Technology

Project

A project assesses the learner's ability to apply knowledge, skills and a range of competencies in an integrated manner, many of which cannot be assessed in other ways. It has a degree of open-endedness, but is focused and results in individual but similar tasks. The time to complete a project normally ranges from a few days to several weeks.

In IT the project is the practical assessment task (PAT).

The project should enable a learner to apply a combination of techniques, knowledge and skills to new situations to complete the task or accomplish a goal. It should also encourage learners to use and combine information, data, and ideas to solve problems, discover and explain patterns, relationships or trends and predict behaviour/events.

A project should require the learner to:

- do some planning/preparation/investigation/research/data gathering to solve the identified problem/task;
- perform the task/carry out instructions (according to criteria given);
- produce a product such as a software application (this could include a limited number of smaller products such as a planning document, that builds up to the final product, which the teacher could monitor or assess informally or formally);
- demonstrate thinking and decision making skills; and
- demonstrate some innovation and creativity.

To set and manage the project, the teacher should:

- determine the content/skills/knowledge to be addressed;
- set clear criteria and give clear instructions to guide the learner (the learner should know exactly what to do and what is expected);
- keep the scope manageable;
- determine which resources will be required to complete the project and ensure that learners have access to these resources;
- determine the time frame/duration/due date;
- determine mark distribution and compile an assessment tool; and
- continuously monitor the completion of the project and guide the learners.

Tests

A test could be a practical test or a written test. The programme of assessment should reflect a balance between practical and written tests. Tests could include open book tests.

A test for formal assessment should not comprise of a series of small tests, but should cover a substantial amount of content and the duration should be 45 – 60 minutes.

Open book tests require learners to find information and apply knowledge and skills. Learners are tested on understanding and application of learning material and not on rewriting. Open book tests should not include only short questions. They must include questions/tasks that will encourage thinking and decision making.

For written open book tests, learners are required to write longer reflective answers, such as paragraph type responses to a given scenario. Paragraphs providing reasons and supporting evidence/arguments are essential.

For practical open book tests learners are required to apply a combination of a series of procedures and techniques to new situations in order to provide a specific answer or accomplish a specific goal.

Alternative Tasks

Integrated test

Integrated test requires learners to be able to apply their knowledge and skills in both theory and practical work that was covered. Testing these types of scenarios e.g. database theory together with database practical, algorithm with implementation and using a trace table to debug a programme.

Case study

Case studies are in-depth investigations of real-life situation. Data is gathered from a variety of sources and by using several different methods. A case study is a research method involving an in-depth, and detailed examination of a scenario, as well as its related contextual conditions.

The duration of a case study should be 45 to 60 minutes

Each test, open book test, alternative task and examination must reflect different cognitive levels.

Formal assessments must cater for a range of cognitive levels and abilities of learners as shown in the table below:

COGNITIVE LEVEL	TAXONOMY	DESCRIPTION
C1	Knowledge, Remembering	Recall of factual/process knowledge **in isolation**, i.e. one step/ set of basic steps/instruction/process at a time, e.g. definitions in the theory paper and known procedures/algorithms in the practical paper.
C2	Understanding, Applying	Demonstrates **understanding** of steps/algorithms/processes/ isolatable bits, such as translating from one form of representation to another, e.g. converting a flow chart representation of a program/program segment to a functional program. It also requires using **known routines/algorithms/processes** in a familiar context in order to complete a task, where **all of the information required is immediately available to the learner**.
C3	Analysing, Evaluating, Creating	Requires **reasoning/investigation/developing a plan** or sequence of steps/algorithm; has some complexity where candidates need to see how parts relate to a whole; organising/ putting together component parts/elements to form a coherent functional whole/achieve an overall objective and completing a task could have more than one possible approach. It could also require weighing possibilities, deciding on the most appropriate solution, as well as testing to locate errors/ troubleshooting, pattern recognition and generalisation. These questions will comprise actions/strategies/procedures where candidates are required to create their own solutions to challenges they may encounter. These questions could include analysing questions or data, and decision-making.

Levels of difficulty are categorised as follows:

D1: Easy for the average Grade 12 candidate to answer

D2: Moderately challenging for the average Grade 12 candidate to answer

D3: Difficult for the average Grade 12 candidate to answer

D4: Very difficult for the average Grade 12 candidate to answer. The skills and knowledge required to answer questions at this level should be included to distinguish amongst high achievers.

A detailed taxonomy for practical:

Lower Order (C1)	**Middle Order (C2)**	**Higher Order (C3)**
30%	**40%**	**30%**
Knowledge/Remembering (Annexure B)	Understanding/applying	Analysing/evaluating/creating
Routine (known Procedures) Use in *isolation*	*Multi*-procedures Combine concepts/isolatable bits	Problem Solving Develop/Create Solution
Code Generator *Operates at level of **individual** lines of code/code structures/ routine procedures (**in isolation)***	***Program Generator*** *Operates at level of writing basic programs that combine concepts/structures, isolatable bits*	***Software Developer*** *Operates at a level of writing solutions to new/unfamiliar or open-ended problems*
The learner is able to • *recall* specific **isolatable** bits of information learned • *use bits of code/code structures **in isolation** - no real connections – in an unrelated way* • *generate code - knows syntax and semantics - can write a line of code/a code structure that does something specific, e.g. basic processing statement, lines of code to obtain input or produce output, algorithm to swop two items, etc.* • *to focus on one relevant aspect at a time (uni-structural)* • *answer questions, seen before, used in **exactly same context** as learned/classroom-based exercise and that is straight forward, to-the-point, that requires mostly one, direct answer/piece of code/code structure* ***Cannot*** • *See relationships* • *Combine concepts/various lines of code/code structures to achieve a goal or complete a task*	*The learner is able to* • *read a program, tell what each line means/ does* • *tell the goal/outcome of a program* • *write programs seen before in a similar context/to perform specific tasks* • *Able to relate, combine and integrate some concepts/code/code structures into valid programs - use and combine specific building blocks to write a program for a specific task* • *Can answer closed/scaffolded questions in a **similar** context than experienced before, with or without new elements* ***Cannot*** • *Optimise a program/code* • *Do detailed planning* • *Perform error catching/trace errors* • *Answer unfamiliar, unseen or open-ended questions without scaffolding and guidance*	*The learner is able to* • *tell what the different parts of a program do and how different parts of a program work together* • *optimise a program/section of code* • *analyse, design, plan, implement and test a solution to a new problem* • *Perform error catching, understanding when, where and how* • *relate, combine and integrate several code structures/constructs to devise 'new' algorithms/ adapting existing ones* • *link several aspects to a broader context* • *independently identify patterns and relate these to programming constructs/structures* • *generalise, abstract and decompose problems into sub-problems and modules* • *answer free response/open-ended questions, 'new' (**unseen** questions), by framing the question and finding a plausible cause of action*

Examples of isolatable bits of content learned (***knowledge)*** that the learner is able to recall and use **in isolation**: • syntax rule • code statement, e.g. assign statement • built-in method, e.g. random • structure, e.g. class definition • algorithm, e.g. swap two values, sort • process, e.g. reading a text file, populate array • setting property value	***Understanding***: Convert from one format to another, e.g. interpret flow chart and convert to code, Read code and tells what it does or provide the output ***Applying***: Carrying out or using a procedure/algorithm/structure/ code statement in a given situation **similar context** (but *new elements or situation*) as was experienced before to perform a task, e.g. combine concepts/isolatable bits	***Synthesis*** Combine concepts in **unfamiliar/new** context to form a (new coherent or functional whole), e.g. code a solution to a problem/to perform a task (not seen before) Includes ***analysing,*** e.g. identifying different parts such as sub-routines/ modules/data structures / I/O strategies /algorithms required; Includes ***evaluating***, e.g. deciding which structures to use and free response/open-ended questions

Questions in the formal assessments will assess performance at different cognitive levels, critical thinking skills, problem-solving techniques and difficulty, as outlined below.

In judging the level of difficulty of each question, both the demands that each question makes on the cognitive ability of an average Grade 12 IT learner and the intrinsic level of difficulty of the question or task are considered. In making this judgement, the difficulty or ease of a particular question is identified. A four-category framework for thinking about question or item difficulty adapted from Leong (2006) has been used in this identification process. This framework comprises the following four general categories of difficulty:

- **Content difficulty:** This indexes the difficulty of the subject matter, topic or conceptual knowledge; some content is inherently more difficult than other content.
- **Stimulus difficulty:** This relates to the linguistic features of the question and the challenge that candidates face in reading, interpreting and understanding the question.
- **Task difficulty:** This refers to the difficulty that candidates face when trying to formulate or produce an answer.
- **Expected response difficulty:** This refers to difficulties because of the mark scheme or marking guidelines, in other words how marks are to be allocated. Therefore, answers to multiple-choice questions on a specific topic could be easier than questions where a candidate has to construct a response.

Weighting of cognitive levels and difficulty levels

Papers 1 and 2 will include questions across three cognitive levels. The distribution of cognitive levels in the practical and theory papers is given in the table below.

COGNITIVE LEVEL	DESCRIPTION	PAPER 1 (PRACTICAL)	PAPER 2 (THEORY)
1	Knowledge and remembering	30%	30%
2	Understanding and applying	40%	40%
3	Analysing, evaluating and creating	30%	30%

The estimated percentages for each level of difficulty within each cognitive level are shown in the table below.

	D1	D2	D3	D4	TOTAL
C1	±10%	±10%	±10%	-	**±30%**
C2	±15%	±15%	±8%	±2%	**±40%**
C3	±15%	±7%	±5%	±3%	**±30%**
TOTAL	**±40%**	**±32%**	**±23%**	**±5%**	**100%**

Learners are required to investigate and analyse problems in a variety of contexts (such as scientific, technological, environmental and everyday-life contexts) in order to solve the described problems effectively, either via programming code in Paper 1 or describe proposed solutions in Paper 1/Paper 2.

4.4 Programme of Assessment

The following tables provide the programme of assessment requirements for Information Technology:

Grade 10

<table>
<tr><td colspan="4">Programme of Assessment</td></tr>
<tr><td colspan="4">SBA per Term</td></tr>
<tr><td>Term 1:
1 Theory Test (100%)</td><td>Term 2:
1 Practical Test (30%)
+
1 Examination comprising
2 Papers:
1 Theory (35%) +
1 Practical (35%)</td><td>Term 3:
1 Practical Test (50%)
+
1 Alternative Task:
Closed or Open Book or Case Study or Integrated test (50%)</td><td>Term 4:
1 Examination comprising
2 Papers:
1 Theory + 1 Practical

PAT</td></tr>
<tr><td colspan="4">Term Mark (Terms 1 – 3):
• Term 1 = 100%
• Term 2 = 30%, 70% (35%, 35%)
• Term 3 = 50%, 50%
Promotion Mark:
• Calculate weighted totals for assessment tasks from term 1 to term 3 and convert to 25%
• Convert PAT mark to 25%
• Convert Paper 1 to 25%
• Convert Paper 2 to 25%</td></tr>
</table>

Grade 11

Programme of Assessment **SBA per Term**			**External Assessment**
Term 1: 1 Practical Test (50%) + 1 Theory Test (50%)	**Term 2:** 1 Examination comprising 2 Papers: 1 Theory (50%) + 1 Practical (50%)	**Term 3:** 1 Practical Test (50%) + 1 Alternative Task: Closed or Open Book or Case Study or Integrated test (50%)	**Term 4:** 1 External Examination comprising 2 Papers: 1 Theory + 1 Practical PAT
		Practical Assessment Task (PAT)	

Term Mark (Terms 1 – 3):

- Term 1 = 50%, 50%
- Term 2 = 50%, 50%
- Term 3 = 50%, 50%

External Examination:

- Calculate weighted totals for assessment tasks from term 1 to term 3 and convert to 25%
- Convert Paper 1 to 25%
- Convert Paper 2 to 25%
- Convert PAT to 25%

Grade 12

Programme of Assessment **SBA per Term**			**External Assessment**
Term 1: 1 Alternative Task: Closed or Open Book or Case Study or Integrated test (50%) + 1 Theory Test (50%)	**Term 2:** 1 Practical Test (30%) + 1 Examination comprising 2 Papers: 1 Theory (35%) + 1 Practical (35%)	**Term 3:** 1 Examination comprising 2 Papers: 1 Theory (50%) + 1 Practical (50%)	**Term 4:** 1 External Examination comprising 2 Papers: 1 Theory + 1 Practical PAT
	Practical Assessment Task (PAT)		

Term Mark (Terms 1 – 3):

- Term 1 = 50%, 50%
- Term 2 = 30%, 70% (35%, 35%)
- Term 3 = 50%, 50%

SBA Mark:

- Calculate weighted totals for assessment tasks from term 1 to term 3 and convert to 25%

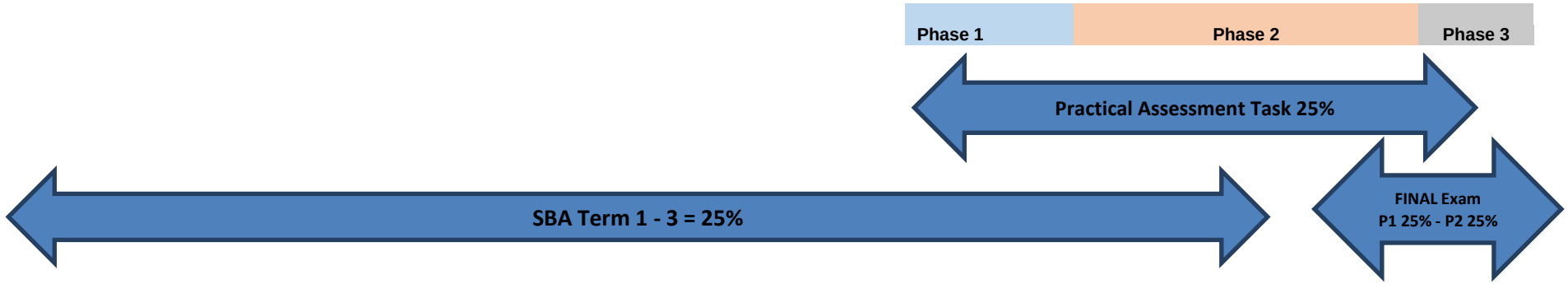
External Examination:

- Convert Paper 1 to 25%
- Convert Paper 2 to 25%
- Convert PAT to 25%

In the table below a more detailed programme of assessment

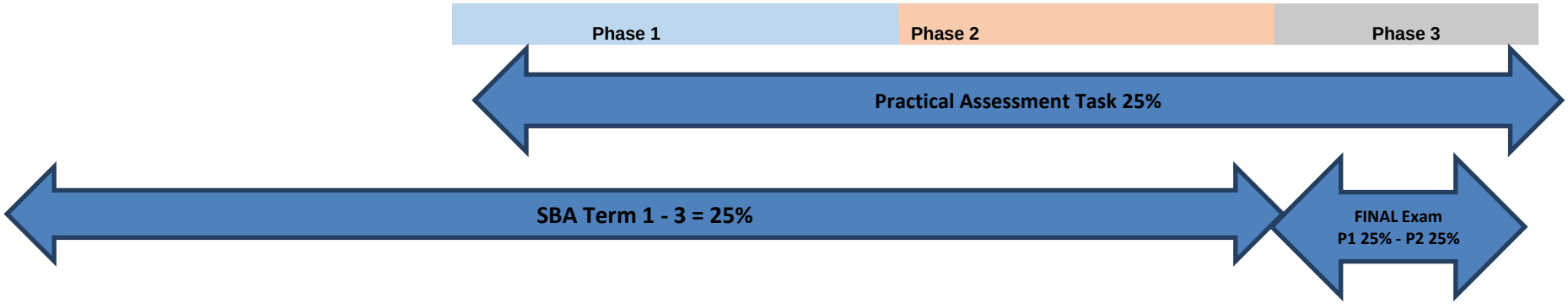
GRADE 10

	TERM 1		TERM 2						TERM 3				TERM 4	
	Task 1	**Weight**	**Task 2**	**Weight**	**Task 3**				**Task 4**	**Weight**	**Task 5**	**Weight**	**Task 6**	
Form / Types of Assessment	Test 1 Theory		Test 2 Practical		Exam Practical		Exam Theory		Alternative Task: Closed or Open Book or Case Study or Integrated test		Test 5 Practical		Final Practical Exams	Final Theory Exams
Tool(s) of Assessment	Question Paper & Memo		Question Paper & Marking Guideline		Question Paper & Marking Guideline		Question Paper & Memo		Question Paper, Memo, Marking Rubric		Question Paper & Marking Guideline		Question Paper & Marking Guideline	Question Paper & Memo
Total Marks	Min 45	**12.0%**	Min 45	**12.0%**	100	**26%**	100	**26%**	Min 45	**12.0%**	Min 45	**12.0%**	120	120
Time Allocation	45 - 60 Minutes		45 - 60 Minutes		2.5 Hours		2 Hours		45 - 60 Minutes		45 - 60 Minutes		3 Hours	2.5 Hours
Date Of Completion	Before end of Term 1		Before end of Term 1		Before end of term 2		Before end of term 2		Before end of Term 3		Before end of Term 3		Before end of term 4	Before end of term 4
Content Focus: Knowledge and Skills	Content covered as per CAPS		Content covered as per CAPS		Content covered as per CAPS		Content covered as per CAPS		Content covered as per CAPS		Content covered as per CAPS		Content covered as per CAPS	Content covered as per CAPS



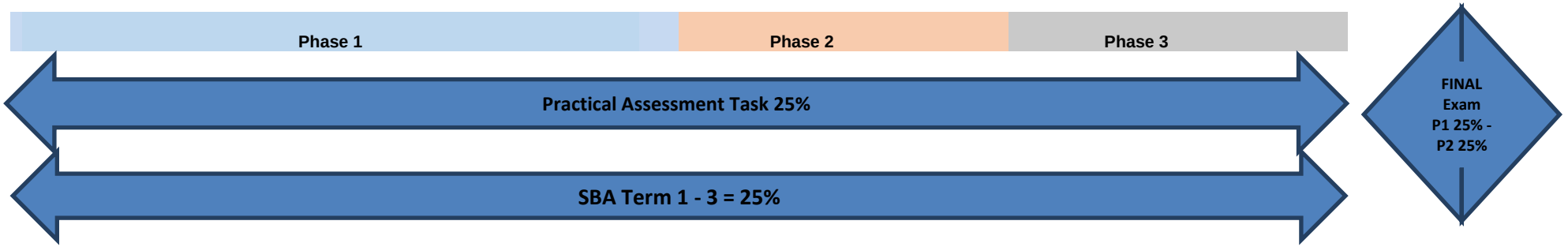
GRADE 11

	TERM 1				TERM 2				TERM 3				TERM 4		
	Task 1	**Weight**	**Task 2**	**Weight**	**Task 3**			**Weight**	**Task 4**		**Task 5**		**Task 6**		
Form / Types of Assessment	Test 1 Theory		Test 2 Practical		Exam Practical		Exam Theory		Alternative Task: Closed or Open Book or Case Study or Integrated tests		Test 5 Practical		Final Practical Exams		Final Theory Exams
Tool(s) of Assessment	Question Paper & Memo		Question Paper & Marking Guideline		Question Paper & Marking Guideline		Question Paper & Memo		Question Paper, Memo, Marking Rubric		Question Paper & Marking Guideline		Question Paper & Marking Guideline		Question Paper & Memo
Total Marks	Min 45	**7.5%**	Min 45	**7.5%**	120	**35%**	120	**35%**	Min 45	**7.5%**	Min 45	**7.5%**	150		150
Time Allocation	45 - 60 Minutes		45 - 60 Minutes		3 Hours		2.5 Hours		45 - 60 Minutes		45 - 60 Minutes		3 Hours		3 Hours
Date Of Completion	Before end of Term 1		Before end of Term 1		Before end of term 2		Before end of term 2		Before end of Term 3		Before end of Term 3		Before end of term 4		Before end of term 4
Content Focus: Knowledge and Skills	Content covered as per CAPS		Content covered as per CAPS		Content covered as per CAPS		Content covered as per CAPS		Content covered as per CAPS		Content covered as per CAPS		Content covered as per CAPS		Content covered as per CAPS



GRADE 12

	TERM 1				TERM 2						TERM 3				TERM 4	
	Task 1	**Weight**	**Task 2**	**Weight**	**Task 3**	**Weight**	**Task 4**			**Weight**	**Task 5**			**Weight**	**FINAL NSC EXAM**	
Form / Types of Assessment	Test 1 Theory		Alternative Task: Closed or Open Book or Case Study or Integrated task		Test 2 Practical		Exam Practical		Exam Theory		Prelim Practical Exam		Prelim Theory Exam		Final Practical Exams	Final Theory Exams
Tool(s) of Assessment	Question Paper & Memo		Question Paper, Memo, Marking Rubric		Question Paper & Marking Guideline		Question Paper & Marking Guideline		Question Paper & Memo		Question Paper & Marking Guideline		Question Paper & Memo		Question Paper & Marking Guideline	Question Paper & Memo
Total Marks	Min 45	**6%**	Min 45	**6%**	Min 45	**6%**	150	**20.5%**	150	**20.5%**	150	**20.5%**	150	**20.5%**	150	150
Time Allocation	45 - 60 Minutes		45 - 60 Minutes		45 - 60 Minutes		3 Hours		3 Hours		3 Hours		3 Hours		3 Hours	3 Hours
Date Of Completion	Before end of Term 1		Before end of Term 1		Before Mid year Exam		Before end of term 2		Before end of term 2		Before end of Term 3		Before end of Term 3		Before end of term 4	Before end of term 4
Content Focus: Knowledge and Skills	Content covered as per CAPS		Database content covered as per CAPS		Content covered as per CAPS		Content covered as per CAPS		Content covered as per CAPS		Content covered as per CAPS		Content covered as per CAPS		Content covered as per CAPS	Content covered as per CAPS



4.4.2 Examinations

Practical Assessment Task (25% of the total marks for the subject)

The IT PAT assesses the learners' ability to develop a solution for a specific task using the software development tools studied in Grades 10 – 12.

Learners should apply appropriate problem-solving techniques and software engineering principles in developing the application.

The IT PAT comprises different components/stages that represent the software development process using any appropriate approach/methodology. Software development activities typically include aspects such as:

- planning (understanding the problem/task and identifying the requirements), not providing a solution;
- design (interface and program design using appropriate design tools and techniques – learners will not be expected to use any specific software design tool); and
- solution for the PAT (coding, testing, implementation and internal documentation.

The above can be implemented in flexible agile methodology.

In Information Technology the PAT counts 25% of the total promotion/certification mark for the subject. It is implemented throughout the school year and should be undertaken as one extended task, which is broken down into different phases or a series of smaller activities.

Each task must include a declaration of authenticity.

In Grade 12, the criteria for the Practical Assessment Task are externally set, internally administered and marked and externally moderated.

The topic of the PAT will be provided to schools each year by the end of the previous year.

Paper 1: 3-hour practical paper of 150 marks (25% of the total marks for the subject)

This will be a practically oriented paper covering questions on Solution Development.

To successfully complete this paper, each learner must have access to his or her own computer in the exam room. Provision needs to be made for sufficient computers to enable the examination to be completed in **2 sittings.**

This paper assesses the practical skills as well as the knowledge and understanding underlying the skills pertaining to Solution Development, i.e. the high-level programming language studied which includes interaction with a database.

The paper does not have an overarching scenario. Each question may have its own scenario.

The paper will comprise questions covering the following broad topics:

- Programming skills
- OOP-programming (including very basic application and basic problem solving skills)
- Integrated data-aware that will also include problem-solving as part of the solution
- Integrated SQL solution that will also include problem-solving as part of the solution
- Problem solving – different levels of higher order skills

Software design techniques, methods and tools such as UML, CRC cards, etc. will not be examined as part of the practical paper.

The learner will not be required to enter large amounts of data. The required data could be retrieved from the data disk or imported from documents such as a text file, or a database table. All GUIs will be provided.

Marks for questions must be allocated towards basic skills, concepts, constructs and problem solving techniques, e.g. application of an iteration structure as part of the solution (correct structure) as well as for the correct use of the structure. The allocation of marks should take into account the time spent on solving, coding and debugging a solution.

A practical information sheet will be provided for practical (Annexure C).

Paper 2: 3-hour written paper of 150 marks (25% of the total marks for the subject)

The paper will cover all theory aspects of all content, concepts and skills of topics, including elements of Solution Development, e.g. algorithmic development, data structures, program design and general programming concepts as well as generic problem solving questions.

The paper does not have an overarching scenario. Each question may have its own scenario.

The following format could be used:

Section		Description
A	**Human Computer Interaction and Social Implications** These topics could be integrated as part of the other sections and will not be a separate section in the paper.	**Short questions (±25 marks)** A range of short questions covering all topics that could include • multiple-choice and • modified true and false.
B		**Systems Technologies (±20 marks)** Questions related to the content, concepts and skills in the Systems Technologies topic.
C		**Communications Technologies and Network Technologies (±25 marks)** Questions related to the content, concepts and skills in the Communication Technologies and Network Technologies topic.
D		**Data and Information Management (±20 marks)** Questions related to the management of data and the concept of information management.
E		**Solution Development (±20 marks)** Questions aligned to the Solution Development topic which assess the knowledge and understanding underlying the concepts and skills in the Solution Development topic.
F		**Integrated Scenario (±40 marks)** This section is based on a single large-scale scenario and assess all the topics.

Software design tools for examination purposes as part of the theory paper are limited to basic flow charts, class diagrams and use case diagrams.

Content to be covered

Assessment addresses the content as set out in this document. Due to the conceptual progression of the content across the grades, content and skills from Grades 10 – 12 will be assessed in the external papers at the end of Grade 12.

Emerging technologies to be covered for examination purposes will be reviewed every three years.

4.5 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 or his readiness to progress or be 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 a number of ways. These include report cards, parents' meetings, school visitations, parent-teacher conferences, phone calls, letters, class or school newsletters, etc. Teachers in all grades report in percentages against the subject.

7 levels of competence have been described for each subject listed for Grades R – 12. The various achievement levels and their corresponding percentage bands are as shown in the table below:

Codes and percentages for recording and reporting

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

Teachers will record actual marks against the task by using a record sheet; and report percentages against the subject on the learners' report cards.

4.6 Moderation of assessment

Moderation refers to the process that ensures that the assessment tasks are fair, valid and reliable. Comprehensive and appropriate moderation practices must be in place for the quality assurance of all subject assessments.

4.6.1 Formal assessment (SBA)

- Grade 10 and 11 tests and examinations are internally moderated. The subject advisor must moderate a sample of these tasks during his/her school visits to verify the standard of tasks and the internal moderation.
- Grade 12 tests and examinations must be moderated at provincial level. This process will be managed by the provincial education department.
- Subject advisors must moderate samples of tests and examination papers before they are written by learners to verify standards and guide teachers on the setting of these tasks.

4.6.2 Practical Assessment Task (PAT)

- Grade 10 and 11: Teachers assess the practical assessment tasks in Grades 10 and 11. The subject advisor must moderate a sample of PATs during his/her school visits to verify the standard of tasks and the internal moderation

- Grade 12: Teachers assess the practical assessment tasks according to the externally set assessment tool. The subject advisor must moderate a sample of each phase of the PATs during his/her school visits to verify the interpretation of the assessment tool and the standard of marking. Completed PATs must also be moderated at provincial level. This process will be managed by the provincial education department.

4.7 Annexures

Annexure A – Glossary of acronyms and abbreviations

Annexure B – Components, events and methods

Annexure C – Practical Information sheet for Grade 12

4.8 General

This document should be read in conjunction with:

4.8.1 *National policy pertaining to the programme and promotion requirements of the National Curriculum Statement Grades R – 12*; and

4.8.2 The policy document, *National Protocol for Assessment Grades R – 12.*

Annexure A

Glossary of Acronyms and Abbreviations

1:M	One-to-many
1-D	One-dimensional
3G	Third generation of cellular wireless
4G	Fourth generation of cellular wireless
ADSL	Asymmetric Digital Subscriber Line
BIOS	Basic Input Output System
CPU	Central Processing Unit
DBA	Database Administrator
DBMS	Database Management System
EDP	Event Driven Programming
ERD	Entity Relationship Diagrams
FOSS	Free Open Source Software
FTP	File Transfer Protocol
GIGO	Garbage-In Garbage-Out
GPS	Global Positioning System
GPU	Graphic Processing Unit
GUI	Graphical User Interface
HAN	Home Area Network
HCI	Human Computer Interface
HDD	Hard Disk Drive
HDMI	High Definition Multimedia Interface
HTML	Hypertext Mark-up Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
I/O	Input-Output
ICT	Information and Communication Technology
IDE	Integrated Development Environment
IP	Internet Protocol
IPO	Input-Processing-Output
IT	Information Technology
JPEG	Joint Photographic Experts Group
LAN	Local Area Network
LTE	Long-Term Evolution
MP3	MPEG-1 Audio Layer-3
MPEG	Motion Picture Expert Group
NFC	Near Field Communication
NIC	Network Interface Card
OOP	Object-Oriented Programming

OS	Operating System
PAN	Personal Area Network
PAT	Practical Assessment Task
PC	Personal Computer
PnP	Plug-and-Play
PoA	Programme of Assessment
POP3	Post Office Protocol
PoS	Point-of-Sale
RAD	Rapid Application Development
RAM	Random Access Memory
RFID	Radio-Frequency Identification
ROM	Read-Only Memory
SaaS	Software as a Service
SEO	Search Engine Optimisation
SMTP	Simple Mail Transfer Protocol
SMS	Short Message System
SOHO	Small Office Home Office
SQL	Structured Query Language
SSD	Solid State Drive
SSL	Secure Socket Layer
TOE	Task-Objects-Events
UML	Unified Modelling Language
URL	Uniform Resource Locater
USB	Universal Serial Bus
VOD	Video On Demand
VoIP	Voice over Internet Protocol
VPN	Virtual Private Network
VRAM	Video Random Access Memory
W3C	World Wide Web Consortium
WAN	Wide Area Network
Wi-Fi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WWW	World Wide Web
WYSIWIG	What You See Is What You Get
XML	Extensible Markup Language

Annexure A
Components, Events and Methods

	Component, Event, Method	10	11	12
Components	TForm	√		
	TPageControl	√		
	TButton	√		
	TBitButton	√		
	TLabel	√		
	TEdit	√		
	TImage	√		
	TPanel	√		
	TListBox	√		
	TRadioGroup	√		
	TComboBox	√		
	TRichEdit	√		
	TRadioButton	√		
	TCheckBox	√		
	TSpinEdit	√		
	TStringGrid		√	
	TDBGrid		√	
	TADOTable		√	
	TADOQuery		√	
	TDataSource		√	
	InputBox()	√		
	ShowMessage()	√		
	MessageDlg()	√		
Events	OnClick()	√		
	OnCreate()	√		
	OnActivate()	√		
	OnShow()	√		
	OnClose()	√		

	Component, Event, Method	10	11	12
Methods - Functions - Procedures	Show()		√	
	Hide()		√	
	Length()	√		
	setLength	√		
	SetFocus()	√		
	Set (IN operator)	√		
	Pos()		√	
	Copy()		√	
	Insert()		√	
	Delete()		√	
	AssignFile()	√		
	Append ()	√		
	Reset()	√		
	Rewrite()	√		
	CloseFile ()	√		
	Ord()		√	
	Chr()		√	
	Val()		√	
	Str()		√	
	UpCase()	√		
	UpperCase()	√		
	LowerCase ()		√	
	IgnoreCase ()		√	
	FileExists()	√		
	Readln()	√		
	Writeln()	√		

	Component, Event, Method	10	11	12
Methods - Functions - Procedures	**Conversion / Formatting**			
	IntToStr () / StrToInt ()	√		
	FloatToStr () / StrToFloat ()	√		
	FloatToStrF ()	√		
	DateTime Functions			
	FormatDateTime()		√	
	TimeToStr()		√	
	DateToStr()		√	
	DateTimeToStr()		√	
	StrToDate()		√	
	StrToTime()		√	
	Now()		√	
	Date()		√	
	Time()		√	
	IsLeapYear()		√	
	Mathematical Methods			
	Random()	√		
	RandomRange()	√		
	Round()	√		
	Trunc()	√		
	Frac()	√		
	Ceil()	√		
	Floor()	√		
	Sqr()	√		
	Sqrt()	√		
	Inc()	√		
	Dec()	√		
	PI	√		

NOTE: All ticked components, events and methods are examinable in Grade 11 and 12

Annexure C

DELPHI INFORMATION SHEET

Function	Purpose
Pos(var: String, var: String)	returns the position of a character in a string
Copy(var String, var: integer, var: integer)	copy a substring , from position, how many characters
Insert(var: String, var: String, var: integer)	insert a Substring, into a String, at a position
Delete(var String, var: integer, var: integer)	delete from a string, from position, how many characters
AssignFile(var: filename,'filename.txt')	assigns the logical file filename to the physical filename
Reset(var: filename)	open file for reading
Rewrite(var: filename)	Erases/creates a new file and opens the file for writing
CloseFile (var : filename)	closes the file
Ord(var: char)	returns the ordinal value of a character
Chr(var: integer)	returns the character represented by the ordinal value
Str(var: integer)	Converts an integer value to a string.