education

Department:
Education
REPUBLIC OF SOUTH AFRICA

NATIONAL CERTIFICATES (VOCATIONAL)

SUBJECT GUIDELINES

INTRODUCTION TO SYSTEMS DEVELOPMENT

NQF LEVEL 2

September 2007

INTRODUCTION

A. What is Introduction to Systems Development?

Introduction to Systems Development provides the student with an awareness of the concepts relating to systems development.

B. Why is Introduction to Systems Development important in the Information Technology and Computer Science programme?

It provides the student with the necessary foundation needed to understand the development of computer programs.

C. The link between the Introduction to Systems Development Learning Outcomes and the Critical and Developmental Outcomes

The student will be able to identify and solve systems development problems faced by the business organisation by collecting, organising, analysing and critically evaluating relevant information. The student will also demonstrate an understanding of the world as a set of interrelated systems by recognising problem-solving contexts do not exist in isolation.

D. Factors that contribute to achieving the Introduction to Systems Development Learning Outcomes

- Analytical and logical ability
- Keen powers of observation
- Ability to transfer skills from familiar to unfamiliar situations
- Meticulous nature
- Interest in computers and related topics

INTRODUCTION TO SYSTEMS DEVELOPMENT – LEVEL 2

CONTENTS

1 DURATION AND TUITION TIME

This is a one-year instructional programme comprising 200 teaching and learning hours. The subject may be offered on a part-time basis provided the student meets all the assessment requirements.

Students with special education needs (LSEN) must be catered for in a way that eliminates barriers to learning.

2 SUBJECT LEVEL FOCUS

The student will be able to explain systems development concepts.

3 ASSESSMENT REQUIREMENTS

3.1 Internal assessment (50 percent)

3.1.1 Theoretical component

The theoretical component forms 60 percent of the internal assessment mark.

Internal assessment of the theoretical component in Introduction to Systems Development Level 2 takes the form of observation, class questions, group work, informal group competitions with rewards, individual discussions with students, class, topic and semester tests and internal examinations. Lecturers can observe students when marking exercises from the previous day and asking class questions.

Assignments, case studies and tests can be completed at the end of a topic. Tests and internal examinations must form part of the internal assessment.

3.1.2 Practical component

The practical component forms 40 percent of the internal assessment mark.

Practical components include applications and exercises. All practical components must be indicated in a Portfolio of Evidence (PoE).

Internal assessment of the practical component in Introduction to Systems Development Level 2 takes the form of assignments, practical exercises, case studies and practical examinations in a simulated business environment.

Students may complete practical exercises daily. Assignments and case studies can be completed at the end of a topic. Practical examinations can form part of internal practical assessment.

- **Some examples of practical assessments include, but are not limited to:**
  A. Presentations (lectures, demonstrations, group discussions and activities, practical work, observation, role-play, independent activity, synthesis and evaluation)
  B. Exhibitions by students
  C. Visits undertaken by students based on a structured assignment task
  D. Research
  E. Task performance in a “Structured Environment”

- **Definition of the term “Structured Environment”**

For the purposes of assessment, “Structured Environment” refers to a simulated workplace or workshop environment. It is advised that a practicum room is available on each campus for practical assessment.

- **Evidence in practical assessments**

All evidence pertaining to evaluation of practical work must be reflected in the students’ Portfolio of Evidence (PoE). The tools and instruments constructed and used to conduct these assessments must be clear from the evidence contained in the Portfolio of Evidence (PoE).

3.1.3 Processing of internal assessment mark for the year

A year mark out of 100 is calculated by adding the marks of the theoretical component (60 percent) and the practical component (40 percent) of the internal continuous assessment (ICASS).

3.1.4 Moderation of internal assessment mark

Internal assessment is subjected to internal and external moderation procedures as set out in the *National Examinations Policy for FET College Programmes*.

3.2 External assessment (50 percent)

A National Examination is conducted annually in October or November by means of a paper(s) set and moderated externally. A practical component will also be assessed.

External assessment details and procedures are set out in the *Assessment Guidelines: Introduction to Systems Development* (Level 2).

4 WEIGHTED VALUES OF TOPICS

TOPICS	WEIGHTED VALUE
1. Basic Concepts of Software	10
2. Software Development and Programming Languages Concepts	10
3. Concepts of Artificial Intelligence	10
4. Computer Data Storage	25
5. Principles of Computer Programming	20
6. Principles of the Internet and Worldwide Web (www)	25
TOTAL	**100**

5 CALCULATION OF FINAL MARK

Internal assessment mark: Student's mark/100 x 50 = a mark out of 50 (a)

Examination mark: Student's mark/100 x 50 = a mark out of 50 (b)

Final mark: (a) + (b) = a mark out of 100

All marks are systematically processed and accurately recorded to be available as hard copy evidence for, amongst others, reporting, moderation and verification purposes.

6 PASS REQUIREMENTS

The student must obtain at least fifty (50) percent in ICASS and fifty (50) percent in the examination.

7 SUBJECT AND LEARNING OUTCOMES

On completion of Introduction to Systems Development Level 2, the student should have covered the following topics:

Topic 1: Basic Concepts of Software
Topic 2: Software Development and Programming Concepts
Topic 3: Concepts of Artificial Intelligence
Topic 4: Computer Data Storage
Topic 5: Principles of Computer Programming
Topic 6: Principles of the Internet and Worldwide Web (www)

7.1 Topic 1: Basic Concepts of Software

Subject Outcome 1: Explain what software is and categorise the types of software.

Learning Outcomes:

The student should be able to:

- Explain the term software.
- Differentiate between the types of software and their purposes.
- Differentiate between basic and advanced software tools.
- Outline the reasons for different versions within the same software.
- Outline the processes for installing application software.

Subject Outcome 2: Describe some features common to all types of application software.

Learning Outcomes:

The student should be able to:

- Identify and demonstrate the different features common to all types of application software.
- Explain the purpose and use of the types of features common to all types of application software.

Subject Outcome 3: Define system software.

Learning Outcomes:

The student should be able to:

- Briefly describe the term system software.
- Define the operating system in terms of the tasks it performs in a computer.
- Define utility programs in terms of their use.
- Define language translators in terms of their purpose, with examples.

Subject Outcome 4: Name and describe microcomputer operating systems and operating environments.

Learning Outcomes:

The student should be able to:

- Name and describe different operating systems.
- Describe the environment in which the operating system operates.
- Outline the history of the different operating systems.

7.2 Topic 2: Software Development and Programming Languages Concepts

Subject Outcome 1: Describe the generations of programming languages

Learning Outcomes:

The student should be able to:

- List the generations of programming languages in which they have evolved.
- Outline programming languages in terms of technicality, flexibility, user-friendliness and speed.
- Compare the strengths and limitations of programming languages.

Subject Outcome 2: Describe the uses for some of the most popular high-level programming languages.

Learning Outcomes:

The student should be able to:

- List and describe the most popular high-level programming languages.
- Explain the uses of high-level programming languages.
- Compare the advantages and disadvantages of high-level programming languages.

Subject Outcome 3: Describe concepts relating to object-oriented and visual programming.

Learning Outcomes:

The student should be able to:

- Describe object-oriented and visual programming concepts.
- Explain object-oriented programming in terms of recycling technique steps.
- Explain object-oriented programming language in terms of the concepts involved.
- List examples of object-oriented programming languages.
- Explain the visual programming language in terms of its goals.

Subject Outcome 4: Name and discuss basic steps in developing a computer program.

Learning Outcomes:

The student should be able to:

- Name the basic steps for developing a computer program.
- Discuss briefly the basic steps involved in a computer program development cycle.

Subject Outcome 5: Describe software development tools

Learning Outcomes:

The student should be able to:

- Name examples of software development tools.
- Briefly describe these software development tools.

7.3 Topic 3: Concepts of Artificial Intelligence

Subject Outcome 1: Define artificial intelligence and how it is used.

Learning Outcomes:

The student should be able to:

- Define the term artificial intelligence.
- Define artificial intelligence in terms of its development of technology.
- List the uses of artificial intelligence in the business environment.

Subject Outcome 2: Describe how robots are used.

Learning Outcomes:

The student should be able to:

- Describe the term robot in simpler terms.
- Describe robots in terms of the past, present and future.
- Describe robots in terms of perception systems.
- Describe the uses of robots.
- Explain the impact of robots on technology.

Subject Outcome 3: Define natural language and fuzzy logic.

Learning Outcomes:

The student should be able to:

- Define the term fuzzy logic.
- Define fuzzy logic in terms of its development.
- List the advantages and disadvantages of fuzzy logic.

Subject Outcome 4: Describe an expert system.

Learning Outcomes:

The student should be able to:

- Explain expert systems.
- List the users of expert systems.
- Describe the major components of an expert system.
- Outline the implications for businesses using expert systems.

Subject Outcome 5: Explain neural networks.

Learning Outcomes:

The student should be able to:

- Explain the term neural network.
- Explain problems addressed by neural networks.
- Explain the categories of neural networks.
- Name and explain the three main parts found in a neuron in neural networks.
- Define the term synapses.

Subject Outcome 6: Define virtual reality and its applications.

Learning Outcomes:

The student should be able to:

- Explain what virtual reality is.
- Describe the applications of virtual reality.

7.4 Topic 4: Computer Data Storage

Subject Outcome 1: Demonstrate an understanding of computer data types.

Learning Outcomes:

The student should be able to:

- Distinguish between data types and their examples.
  Range: Bits (0,1), bytes, numbers , integers (+- values, whole items), floating point (temperature, voltage, etc) and Boolean (0/1 on/off)
- Distinguish categories of coding systems and their uses in a business environment.
  Range: BCD, EBCDIC (IBM mainframes), ASCII (micro and mini computers and, UNICODE
- Explain and illustrate how data manipulation operations are performed on data types.

Subject Outcome 2: Describe computer data structures.

Learning Outcomes:

The student should be able to:

- List examples of computer data structure types.
- Distinguish types of computer data structure.
  Range: Bits, bytes, characters, fields, records, files and databases
- Distinguish types of computer file and identify their examples.
  Range: Master, transaction, temporary, document, serial, sequential, indexed and direct
- Distinguish types of computer database.
- List examples of computer databases.
  Range: Hierarchical, network, relational and SQL (completeness, non-redundancy, structure)

7.5 Topic 5: Principles of Computer Programming

Subject Outcome 1: Describe problem analysis and program design techniques.

Learning Outcomes:

The student should be able to:

- Name and describe the steps and techniques of program maintenance.
- Identify different problem analysis techniques.
- Identify different programming design techniques.

Subject Outcome 2: Describe the different data representations used in computer programs.

Learning Outcomes:

The student should be able to:

- Distinguish between different numeric data types.
- Describe the different logical data types.
- Distinguish between different internal representations of data types.
- Describe the different logical operators.

Subject Outcome 3: Describe the basic principles of computer programming.

Learning Outcomes:

The student should be able to:

- Describe the different algorithmic structures of programming languages.
- Explain the principles of good program documentation.
- Explain the principles of programming quality assurance (QA).
- Distinguish between validation and verification.
- Explain the relationship between file, records and fields.

Subject Outcome 4: Describe the principles used in designing a computer program.

Learning Outcomes:

The student should be able to:

- Explain the methods of specifying problems.
- Explain the techniques used to research problems in terms of inputs and outputs.
- Explain how to evaluate the viability of developing computer programs to solve problems.
- Identify the issues involved in assessing the viability of developing computer programs.
- Explain the features of a computer program that can solve a given problem.

7.6 Topic 6: Principles of the Internet and Worldwide Web (www)

Subject Outcome 1: Explain the principles of the Internet and worldwide web.

Learning Outcomes:

The student should be able to:

- Outline the origin and history of the Internet.
- Identify the major applications of the Internet.
- Explain the use of major Internet applications.
- Explain the history and development of the worldwide web.

Subject Outcome 2: Explain how the worldwide web incorporates various Internet applications.

Learning Outcomes:

The student should be able to:

- Explain in detail the physical context of web pages.
- Explain how the worldwide web can be applied in an intranet and extranet.
- Explain the latest Internet applications, including web-based e-mails, instant messaging and Voice over IP (VoIP).

8 RESOURCE NEEDS FOR THE TEACHING OF INTRODUCTION TO SYSTEMS DEVELOPMENT – LEVEL 2

8.1 Physical resources

The following teaching aids should be made available, if possible:

- Lecture room
- Educator's computer connected to the Internet
- Data projector
- Electronic white board
- Networked computer room or library with Internet access and referencing software, for example Encarta Encyclopaedia
- Networked laser printer per five computer.

8.2 Human resources

- The lecturer must have completed an Information Technology-related subject at NQF Level 5 and be trained in outcomes-based education.
- It will an advantage if the lecturer has already been declared competent as assessor and/or moderator.

8.3 Other resources

- File per student for the Portfolio of Evidence (PoE) and 30 printing sheets of paper per student
- Computer-related books for referencing purposes