NATIONAL CERTIFICATE (VOCATIONAL)

SUBJECT GUIDELINES

COMPUTER PROGRAMMING

NQF Level 4

Implementation: January 2020
INTRODUCTION
1 DURATION AND TUITION TIME
2 SUBJECT LEVEL FOCUS
3 ASSESSMENT REQUIREMENTS
   3.1 Internal assessment
   3.2 External assessment
4 WEIGHTED VALUES OF TOPICS
5 CALCULATION OF FINAL MARK
6 PASS REQUIREMENTS
7 SUBJECT AND LEARNING OUTCOMES
   7.1 Object Oriented Programming
   7.2 Advance data structures
   7.3 Database application design
   7.4 The Internet and world Wide Web application
   7.5 Design and develop a web-site using HTML and CSS
   7.6 Web multimedia and interactivity
8 RESOURCE NEEDS FOR THE TEACHING OF COMPUTER PROGRAMMING - LEVEL 4
   8.1 Physical resources
   8.2 Human resources
   8.3 Other resources
INTRODUCTION

A. What is Computer Programming?

Computer Programming involves the designing and programming of well-tested and user-friendly computer-based solutions to meet specific requirements.

B. Why is Principles of Computer Programming important in the Information Technology programme?

Principles of Computer Programming is important in the Information Technology programme as it enhances the development of the IT environment.

C. The link between the Learning Outcomes for Principles of Computer Programming and the Critical and Developmental Outcomes

The student will be able to identify and solve problems, and collect, analyse, organise and critically evaluate information that is related to computer programming. The student will also demonstrate an understanding of the world as a set of related systems by recognising that problem-solving contexts do not exist in isolation.

D. Factors that contribute to achieving Computer Programming Learning Outcomes

- Analytical and logical ability
- Keen powers of observation
- Transferring of skills from familiar to unfamiliar situations
- Meticulous attention to detail
- Interest in computers and related topics

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 candidate meets all of the assessment requirements.

Course preparation should take consideration of students with special education needs.

2 SUBJECT LEVEL FOCUS

- Apply computer-programming skills.

3 ASSESSMENT REQUIREMENTS

3.1 Internal assessment (50 percent)

3.1.1 Theoretical Component

The theoretical component will form 30 percent of internal assessment.

Internal assessment of the theoretical component of Computer Programming Level 4 will take the form of observation, class questions, group work, (informal group competitions with rewards), individual discussions with students, class, topics and semester tests, internal examinations. Daily observation can be made when marking exercises of the previous day and class questions. Assignments, case studies and tests can be done at the end of a topic. Tests and internal examinations must form part of internal assessment.

3.1.2 Practical Component

The practical component includes applications and exercises. All practical components must be indicated in a Portfolio of Evidence (PoE)

The practical component will form 70 percent of internal assessment.

Internal assessment of the practical component of Computer Programming Level 4 will take the form of assignments, practical exercises, case studies, practical examination in a simulated business environment.
Students may complete practical exercises on a daily basis. Assignments and case studies can be done at the end of a topic. Practical examination can form part of internal assessment.

**Some examples of practical assessments include, but are not limited to:**

- Presentations (lectures, demonstrations, group discussions and activities, practical work, observation, role play, self activity, judging and evaluation)
- Use of aids
- Exhibitions
- Visits
- Guest speaker presentations
- Research
- Task performance in a simulated/structured environment

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

“Structured environment” for the purposes of assessment refers to an actual or 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 student's Portfolio of Evidence. The tools and instruments constructed and used for the purpose of conducting such assessments must be clear from evidence contained in the 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 and the practical component of the internal continuous assessment.

3.1.4 Moderation of internal assessment mark

Internal assessment is subjected to both internal and external moderation procedures as contained 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 set externally and marked and moderated internally.

Details in respect of external assessment are contained in the *Assessment Guidelines: Computer Programming (Level 4).*

4 WEIGHTED VALUES OF TOPICS

<table>
<thead>
<tr>
<th>TOPICS</th>
<th>WEIGHTED VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Object Oriented Programming</td>
<td>25%</td>
</tr>
<tr>
<td>2. Advanced data structures</td>
<td>10%</td>
</tr>
<tr>
<td>3. Database application design</td>
<td>25%</td>
</tr>
<tr>
<td>4. The Internet and World Wide Web application</td>
<td>10%</td>
</tr>
<tr>
<td>5. Design and develop a Website using HTML and CSS</td>
<td>20%</td>
</tr>
<tr>
<td>6. Web multimedia and interactivity</td>
<td>10%</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td>100</td>
</tr>
</tbody>
</table>

5 CALCULATION OF FINAL MARK

Continuous assessment:

Student's mark/100 x 50/1 = a mark out of 50  (a)

Theoretical examination mark:  Student's mark/100 x 50/1= 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, purposes of moderation and verification, as well as purposes of reporting.

6 PASS REQUIREMENTS

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

7 SUBJECT AND LEARNING OUTCOMES

On completion of Computer Programming Level 4 the student should have covered the following topics:

Topic 1: Object Oriented Programming
Topic 2: Advanced data structures.
Topic 3: Data base application design
Topic 4: The Internet and World Wide Web application
Topic 5: Design and develop a website using HTML and CSS
Topic 6: Web multimedia and interactivity

Topic 1: Object Oriented Programming

Subject Outcome 1.1: Describe basic object oriented concepts

Learning Outcomes
The student should be able to:
- Explain the basic principles of a class.
- Explain the basic principles of an object.
- Describe an abstract data type in the context of class data members.
- Identify and differentiate between types of methods in a class.
- Explain the basic principles of information hiding and encapsulation.
- Explain the basic principles of inheritance.
- Explain the basic principles of polymorphism.

Subject Outcome 1.2: Describe the fundamental differences between object oriented and procedural programming

Learning Outcomes
The student should be able to:
- Explain the implementation of classes in object oriented programming, using examples
- Explain encapsulation of data and functions/methods (in classes)
- Describe how global data sharing is minimised to enable weak coupling.
- Describe how modules exhibit functional cohesion.
- Explain and identify possible classes for simple examples and problems

Subject Outcome 1.3: Implement object oriented techniques in development of a solution

Learning Outcomes
The student should be able to:
- Represent a class in a UML diagram
- Identify objects and classes.
- Implement objects and classes
- Implement a class/ classes in a solution to a given problem/ scenario
  Range: object identification, class design, objects instantiation, methods, encapsulation, inheritance, polymorphism

Topic 2: Advanced Data Structures
Subject Outcome 2.1: Demonstrate an understanding of multi-dimensional data structures

Learning Outcomes
The student should be able to:
• Identify and describe the concept of multi-dimensional data structures.
• Explain and illustrate features of the multi-dimensional data structures.
• Explain how to use multi-dimensional data structures when demonstrating the manipulation of data.

Subject Outcome 2.2: Demonstrate an understanding of Object Lists.

Learning Outcomes
The student should be able to:
• Identify and describe the concept of lists of objects.
• Explain and illustrate uses of the arrays of objects.
• Explain how to use object collections and lists when demonstrating the manipulation of data.
  Range: object collection/list: inserting, deleting, modifying

Topic 3: Database application design

Subject Outcome 3.1: Review the requirements for database access for a computer programming solution.

Learning Outcomes
The student should be able to:
• Identify and explain the feasibility of the requirements.
• Explain and identify database access objectives and critical performance factors • Explain how to estimate costs for the development effort required.
• Explain the need for adopting a review procedure to ensure that the outcomes meet the database access requirements.

Subject Outcome 3.2: Design database access for a computer application

Learning Outcomes
The student should be able to:
• Describe and demonstrate design implementation per user requirements
• Describe and demonstrate the design of the database structure that resembles the output from the data analysis
• Describe and demonstrate how to ensure that the structure of each table in the database adheres to the third normal form
• Identify and demonstrate the methods of accessing data.
• Identify the key relationships between the tables within the database
• Explain and demonstrate how to ensure that the data types for primary and foreign keys are consistent throughout the database.

Subject Outcome 3.3: Write program code for database access for an application implementing SQL

Learning Outcomes: The student should be able to:
• Identify and implement a method for external data connection and access using program code
• Demonstrate how the program code will use language constructs to facilitate the implementation of the solution.
• Explain and demonstrate how to join tables in a query to satisfy a requirement
• Describe how to construct program code that preserves the integrity of data being accessed by multiple users and processes.

Subject Outcome 3.4: Test programs for an application that accesses a database

Learning Outcomes:
The student should be able to:
• Describe and demonstrate how testing checks all program logic paths.
• Describe and demonstrate how testing corrects program code to eliminate errors identified.
• Describe and demonstrate how testing verifies that the database access functions in the required environment
Subject Outcome 3.5: Document programs for a computer application that accesses a database

Learning Outcomes:
The student should be able to:
- Describe how the documentation enhances the understanding of the program code.
- Describe how the documentation complements the self-documenting attributes of the program code.
- Develop documentation to support the design, program code and solution.

Topic 4: The Internet and the World Wide Web applications

Subject Outcome 4.1: Explain the internet and the World Wide Web in relation to application development

Learning Outcomes:
The student should be able to:
- Explain the internet and the Web.
- Explain Web accessibility standards for information on the Web.
- Discuss popular uses of the Web.
- Explain the process of Web development.

Subject Outcome 4.2: Explain Networking concepts related to internet applications

Learning Outcomes:
The student should be able to:
- Explain networking models.
- Explain internet protocols.
- Use uniform resource identifiers and domain names.

Subject Outcome 4.3: Assess information on the Web

Learning Outcome:
The student should be able to:
- Explain Reliability of information on the Web.
- Explain the concept of using information ethically on the web.
- Explain copyright issues related to internet development.
- Explain ownership issues related to internet development.
- Explain royalty issues related to internet development.
- Explain version control issues related to internet development.
- Explain security issues related to internet development.
- Explain ways of managing security issues related to internet development.

Topic 5: Design and build a web-site using HTML and CSS

Subject Outcome 5.1: Explain the fundamentals of Web design

Learning Outcomes:
The student should be able to:
- Identify the target audience of a given Website.
- Explain the principles of visual design in a Web environment.
- Write content for the Web to be used in a Website.
- Explain the use of colour for the Web.
- Explain the use of graphics and multimedia for the Web.
- Design navigation for a given Website.

Subject Outcome 5.2: Create basic HTML elements on a web page

Learning Outcomes:
The student should be able to:
• Discuss the advantages and disadvantages of HTML editors and other web-site design tools.
• Explain HTML tags as the basis of HTML development.
• Create HTML documents to develop Web pages.
• Add text elements to Web pages.
• Use list elements to organise data in a Web page.
• Represent information in tables.
• Create hyperlinks using the anchor element.
• Use the div element to group Web content.

Subject Outcome 5.3: Configure colour and text using CSS

Learning Outcomes
The student should be able to:
• Explain CSS attribute-value pair.
• Add CSS to a web page as a presentation layer.
• Configure colour on the Web page using CSS.
• Configure text on the Web page using CSS.
• Use different CSS selectors to select elements.

Subject Outcome 5.4: Add visual elements and graphics to Web pages

Learning Outcomes
The student should be able to:
• Add image elements to Web pages.
• Use background images to style Web pages
• Add a favourites icon to a Web page
• Create visual effects using CSS

Subject Outcome 5.5: Create page layout using HTML and CSS

Learning Outcomes
The student should be able to:
• Implement the CSS box model for an element.
• Control the flow of elements on a Web page
• Position elements on a Web page
• Create menus using links and hyperlinks
• Use HTML structural elements to structure a Web page

Topic 6: Web Multimedia and interactivity

Subject Outcome 6.1: Incorporate Web multimedia and interactivity

Learning Outcomes
The student should be able to:
• Explain multimedia elements in the context of the Web.
• Add audio to a Web page using the audio element.
• Add video to a Web page using the video element.
• Implement interactivity using CSS.
  Range: CSS transform, CSS transition

Subject Outcome 6.2: Get user input using forms

Learning Outcomes
The student should be able to:
• Create a form element on a Web page.
• Add input controls on a form.
• Add a list element on a form.
• Control accessibility on a form.
  Range: Label, fieldset, tabindex
• Style a form using CSS.

Subject Outcome 6.3: Write basic JavaScript code
Learning Outcomes
The student should be able to:
• Explain the popular uses of JavaScript.
• Add JavaScript code to a Web page.
• Access page element using the DOM.
• Use basic programming concepts to perform basic functions.

8 RESOURCE NEEDS FOR THE TEACHING OF COMPUTER PROGRAMMING - LEVEL 4

8.1 Physical resources
The following teaching aids should be made available, if possible:
• Lecture room
• Computer for facilitator
• Overhead projector
• Networked computer laboratory with internet access

8.2 Human resources
• The facilitator must have as a major subject Computer Programming at NQF Level 5.
• It will be to the advantage of facilitator if they have already been declared competent as assessor and/or moderator.
• Training in OBE

8.3 Other resources
• Computer per learner
• Networked laser printer
• SQL software
• Programming software
• Web page design software
• Multimedia software
• Animation software
• File per learner for PoE
• DVD-RW per learner
• Ream of paper per learner
• 1 GB flash disk per learner
• Programming magazines and journals
• Multimedia magazines