



## Department of Computing

### Course: Higher Diploma in Computing

### Module Title: Computer Systems

<b>Credits:</b>	10
<b>Credit Level:</b>	8
<b>Prerequisite Modules:</b>	None

#### *Description:*

The aim of this module is to provide the student with a firm foundation in the principles and practicalities of computer systems: processor and system architecture, characteristics and interactions.

#### *Module Learning Outcomes:*

On successful completion of this module the learner will be able to:

1. Appraise modern computer systems in terms of architecture, operating system and communications capability.
2. Explain how programmes and data are represented and executed.
3. Describe and illustrate the relationship between underlying hardware and the useful features of an operating system.
4. Describe the principle components of an operating system and detail how each component carries out its allotted function and relates to other components.
5. Understand and interpret network models and protocols.
6. Envisage and devise the requirements for infrastructure in a variety of scenarios.

#### *Indicative Content:*

1. Computer Architecture & Organisation
2. Programme Representation
3. Operating Systems
4. Networking

#### *Module Assessment:*

Coursework

100%

**Department of Computing****Course: Higher Diploma in Computing****Module Title: Database Technology**

<b>Credits:</b>	10
<b>Credit Level:</b>	8
<b>Prerequisite Modules:</b>	None

*Description:*

The aim of this module is to allow the student to examine the database environment and to teach them how to design and implement a relational database. In addition the student will learn how to manipulate and define a database using database languages.

*Module Learning Outcomes:*

On successful completion of this module the learner will be able to:

1. Construct an entity relationship model.
2. Map an entity relationship model to a corresponding set of relations and implement the database in a Relational environment.
3. Comprehend the relational integrity rules.
4. Apply the technique of normalisation.
5. Write SQL commands for data manipulation and data definition of a relational schema.
6. Evaluate the appropriateness of a database program to a given business requirement.

*Indicative Content:*

1. The Database Approach
2. Database Design
3. The Relational Model
4. Normalisation
5. SQL
6. SQL Programming

*Module Assessment:*

Coursework

100%

**Department of Computing****Course: Higher Diploma in Computing****Module Title: Software Development**

<b>Credits:</b>	10
<b>Credit Level:</b>	8
<b>Prerequisite Modules:</b>	None

*Description:*

To equip the learner with skills to design and develop object oriented applications.

*Module Learning Outcomes:*

On successful completion of this module the learner will be able to:

1. Design, implement and test object oriented programs.
2. Construct a repository of reusable classes.
3. Differentiate between method overloading and method overriding.
4. Implement programs using arrays and ArrayLists.
5. Browse API documentation to locate and use appropriate classes.
6. Design, implement and test a class hierarchy that implements inheritance and polymorphism.

*Indicative Content:*

1. Overview of Programming Languages
2. Programming Fundamentals
3. Object Oriented Programming
4. Introduction to Data Structures

*Module Assessment:*

*Coursework* *100%*

**Department of Computing****Course: Higher Diploma in Computing****Module Title: Software Engineering and Quality Assurance**

<b>Credits:</b>	10
<b>Credit Level:</b>	8
<b>Prerequisite Modules:</b>	None

*Description:*

This module is designed to equip the student with a firm foundation in software engineering processes, Principles and practices with an emphasis on quality assurance

*Module Learning Outcomes:*

On successful completion of this module the learner will be able to:

1. Compare software process models and justify model selection
2. Devise a project design and development platform and plan its implementation
3. Perform risk assessment in relation to quality assurance
4. Describe Software Quality Management and Software Process Improvement.
5. Utilise testing terminology and QA

*Indicative Content:*

1. Software Engineering & Software Process Models
2. Requirement and Use-case Modelling
3. Project planning & Risk Management
4. Software Quality Management & Testing

*Module Assessment:*

Coursework 100%



## **Department of Computing**

### **Course: Higher Diploma in Computing**

### **Module Title: Internet Applications Development**

<b>Credits:</b>	10
<b>Credit Level:</b>	8
<b>Prerequisite Modules:</b>	None

#### *Description:*

This module will equip learners with the skills needed to plan, design, develop and publish a dynamic database driven web application. The learner will learn about navigation structures, information architecture, usability and accessibility and they will apply these techniques when developing web sites.

#### *Module Learning Outcomes:*

On successful completion of this module the learner will be able to:

1. Create standards-compliant web pages using HTML/CSS.
2. Apply web design principles when developing a web site.
3. Discuss issues relating to information architecture, usability and accessibility.
4. Evaluate and select appropriate technologies for web development projects.
5. Describe the process of serving dynamic web sites.
6. Plan, design and develop dynamic, database driven web sites.

#### *Indicative Content:*

1. HTML
2. CSS
3. Web Design
4. Dynamic Web Development
5. Web Publishing

#### *Module Assessment:*

Coursework 100%

**Department of Computing****Course: Higher Diploma in Computing****Module Title: Software Development & Testing**

<b>Credits:</b>	10
<b>Credit Level:</b>	8
<b>Prerequisite Modules:</b>	None

*Description:*

To enhance the learner's skills in design, implementation and testing of OO applications.

*Module Learning Outcomes:*

On successful completion of this module the learner will be able to:

1. Build graphic user interfaces using the GUI toolkit.
2. Describe, distinguish and implement code using appropriate data structures.
3. Select and apply appropriate algorithms for manipulating data in data structures.
4. Implement programs using input/output, file handling.
5. Implement code using appropriate exception handling.
6. Design and implement appropriate testing strategies for classes.

*Indicative Content:*

1. Input/Output and Exception Handling
2. Data Structures & Algorithms
3. GUI Programming
4. Software Testing

*Module Assessment:*

Coursework 100%