COMPETENCY BASED CURRICULUM

FOR

INFORMATION COMMUNICATION TECHNOLOGY

KNQF LEVEL 5

PROGRAMME ISCED CODE: 061 2454A

COMPUTER PROGRAMMING PRINCIPLES

UNIT CODE: 0613 451 05A

Duration of Unit: 180 Hours

Relationship to Occupational Standards

This unit addresses the Unit of Competency: Apply Computer Programming Principles

Unit Description

This unit covers the competencies required to apply computer programming principles. It involves applying computer programming skills, demonstrating structured programming skills and demonstrating object-oriented programming skills.

Summary of Learning Outcomes

LEARNING OUTCOMES	DURATION (HOURS)
1. Apply Computer programming skills	50
2. Demonstrate Structured programming skills	60
3. Demonstrate Object-oriented programming skills	70
TOTAL	**180**

Learning Outcomes, Content and Suggested Assessment Methods

Learning Outcome	Content	Suggested Assessment Methods
1. Apply computer programming skills	1.1 Identification of Programming Languages 1.1.1 Overview of programming language categories (e.g., procedural, object-oriented, functional)	• Practical Activities • Project work • Demonstration • Group discussions • Observation • Portfolio of

	1.1.2 Criteria for selecting languages based on user requirements 1.2 Application Programming Paradigms 1.2.1.1 Explanation of common programming paradigms 1.2.1.2 Functional 1.2.1.3 Procedural 1.2.1.4 Object-oriented 1.2.1.5 Imperative 1.2.1.6 Declarative 1.2.2 Choosing the appropriate paradigm based on project needs 1.3 Program Development Life Cycle 1.3.1 Stages of the program development life cycle 1.3.2 Best practices for adapting the life cycle to work requirements 1.4 Application of Program Design Tools 1.4.1 Overview of design tools 1.4.1.1 Flow charts 1.4.1.2 Decision tables 1.4.1.3 Decision trees 1.4.1.4 Pseudocode 1.4.1.5 Algorithm 1.4.2 Selecting design tools based on user requirements and project complexity 1.5 Identification of Program Writing Tools 1.5.1 Common program writing tools and IDEs 1.5.1.1 Text editors 1.5.1.2 Compilers Linkers 1.5.1.3 Debuggers	Evidence - Written tests

	1.5.1.4 Special Integrated Development Environment (IDE) 1.5.1 Evaluating tools based on system requirements and developer preferences	
2. Demonstrate structured programming skills	2.1 Declaration of Identifiers in C language 2.1.1 Guidelines for naming conventions and best practices 2.1.2 Ensuring identifiers align with program design specifications 2.2 Initializing Variables and Constants in C language 2.2.1 Importance of proper initialization in programming 2.2.2 Techniques for initialization based on design specifications 2.3 Applying User-Defined Data Types in C language 2.3.1 Overview of user-defined data types in C language 2.3.1.1 Structures 2.3.1.2 Classes 2.3.1.3 Arrays 2.3.1.4 Function 2.3.2 Criteria for selecting data types based on system requirements 2.4 Creating Computer program input in C language 2.5 Application of Data control structures in C program 2.5.1 Types of control structures 2.5.1.1 Selection	● Practical Activities ● Project work ● Demonstration ● Group discussions ● Observation ● Third Party report ● Portfolio of Evidence ● Written tests

	2.5.1.2 Loops 2.5.1.3 Sequence 2.5.2 Best practices for implementing control structures as per design requirements 2.6 Data structures in C program 2.1.1 Overview of common data structures. 2.1.1.1 Arrays 2.1.1.2 Queue 2.1.1.3 Stack 2.1.1.4 Linked lists 2.1.2 Selecting appropriate data structures based on design specifications. 2.2 Creating C computer program subroutines 2.7.1 Benefits of using subroutines 2.7.2 Designing subroutines to meet user needs 2.7.3 Functions and subprograms 2.8 Coding of C Computer program output 2.9 Performing C Computer Program Debugging 2.9.1 Common debugging techniques and tools 2.9.2 Following work procedures for systematic debugging 2.10 Compiling C Computer Program 2.10.1 Steps involved in the compilation process 2.10.2 Ensuring compliance with system requirements during compilation	

3. Demonstrate object-oriented programming skills	3.1 Implementing Objects and Classes in C++ language 3.1.1 Overview of objects and classes in OOP 3.1.2 Ensuring implementation aligns with work procedures 3.2 Declaring Object Methods in C++ language 3.2.1 Defining methods that fulfill application requirements 3.2.2 Best practices for method naming and functionality 3.3 Applying Namespaces in C++ language 3.3.1 Understanding the role of namespaces in OOP 3.3.2 Implementing namespaces 3.4 Data abstraction concepts in C++ language 3.4.1 Definition of data abstraction 3.4.2 Importance of data abstraction 3.4.3 Implementing of data abstraction in OOP 3.5 Object encapsulations in C++ language 3.5.1 Definition of Object encapsulations 3.5.2 Importance of Object encapsulations 3.5.3 Implementing of Object encapsulations in OOP 3.6 Class templates implementation 3.7 Class inheritance implementation 3.7.1 Definition of data abstraction 3.7.2 Importance of data abstraction 3.7.3 Base class 3.7.4 Derived class	• Practical Activities • Project work • Demonstration • Group discussions • Observation • Third Party report • Portfolio of Evidence • Written tests

	3.7.5 Inheritance relationships 3.7.6 Types of inheritance 3.8 Implementing class polymorphism in C++ language 3.8.1 Definition of data polymorphism 3.8.2 Importance of data polymorphism 3.8.3 Implementing of data polymorphism in OOP	

Suggested Delivery Methods

- Instructor led facilitation using active learning strategies
- Demonstration by trainer
- Practical work by trainee
- Viewing of related videos
- Group discussions
- Direct instructions

Recommended Resources for 25 Trainees

S/No.	Category/Item	Description/ Specifications	Quantity	Recommended Ratio (Trainee: Item)
A	**Learning Materials**			
1.	Textbooks	For trainers' use	5 pcs	1:5
2.	Installation manuals	For trainers' use		
3.	Charts	For trainers' use		
4.	PowerPoint presentations	For trainer's use		
5.	Assorted colour of whiteboard markers	For trainer's use		

6.	e-Didactics	For trainer’s use		
B	**Learning Facilities & infrastructure**			
7.	Lecture/theory room	For training	1	1:25
8.	Computer Laboratory	For training	1	1:25
C	**Consumable materials**			
9.	Printing Papers	For printing	1 ream	1:20
10.	Toners	For printers	2 pcs	13: 1
11.	Internet connection	For both trainers’ & trainees’ use		
D	**Tools and Equipment**			
12.	Projectors	For trainers’ use	1	25:1
13.	Printers	For training	4	6:1
14.	Flash drives	For training	5 pcs	5:1
15.	Computers	For training	25 pcs	1:1
16.	Integrated Development Environment (IDEs) – C,C++, Java and Visual Studio, IntelliJ IDEA, Python IDE	For training	25 pcs	1:1