REPUBLIC OF KENYA

COMPETENCY BASED MODULAR CURRICULUM

FOR

COMPUTER SCIENCE

KNQF LEVEL 6

(CYCLE 3)

PROGRAMME ISCED CODE: 0613 554 A.

TVET CDACC
P.O. BOX 15745-00100
NAIROBI

FUNDAMENTALS OF PROGRAMMING

ISCED UNIT CODE: 0613 554 08A

UNIT CODE: ICT/CU/CS/CR/04/6/MA

Relationship to Occupational Standards

This unit addresses the unit of competency: Apply Fundamentals of Programming

Duration of Unit: 200 hours

Unit Description

This unit covers the competencies required to understand fundamentals of programming. It involves understanding programming concepts, understanding the Java environment, performing data operations, using control structures, using methods and understanding Object Oriented programming.

Summary of Learning Outcomes:

Learning Outcomes	Duration (hours)
1. Programming Concepts	20
2. Java environment	30
3. Data operations	35
4. Control Structures	35
5. Use methods	40
6. Object Oriented Programming	40
TOTAL	**200**

Learning Outcomes, Content and Suggested Assessment Methods

Learning Outcome	Content	Suggested Assessment Methods

1. Programming Concepts	1.1 Concepts of programming 1.2 Phases of program development	• Practical tests • Oral tests
	1.2.1 Establish program requirements 1.2.2 Design a program 1.2.3 Coding 1.2.4 Code test and debug 1.2.5 Document 1.2.6 Maintain 1.3 Key terms used in programming 1.3.1 Algorithm 1.3.2 Source code 1.3.3 Executable 1.3.4 Compiling 1.3.5 Debugging 1.4 Types of code 1.4.1 Source code 1.4.2 Object code 1.4.3 Machine code 1.5 Translators used in programming 1.5.1 Compiler 1.5.2 Interpreter 1.5.3 Assembler 1.6 OOP fundamental concepts	• Written tests

2. Java Environment	2.1 Installation of Java 2.1.1 Download Java for Windows 2.1.2 Install JDK 2.1.3 Set the Environment variables 2.2 Java Programming environment 2.2.1 Downloading Eclipse IDE 2.2.2 Setting up Eclipse IDE 2.2.3 Launching Eclipse IDE 2.3 Features of Java 2.4 Java syntax 2.4.1 Case Sensitivity	• Practical tests • Oral tests • Written tests

	2.4.2 Class names 2.4.3 Method names 2.4.4 Program file name 2.4.5 Public static void main 2.4.6 Identifiers 2.4.7 Modifiers 2.4.8 Variables 2.4.9 Java Arrays 2.4.10 Java Enums 2.4.11 Java Keywords	

3. Data Operations	3.1 Java Data Types 3.1.1 Integer 3.1.2 Float 3.1.3 Strings 3.1.4 Boolean 3.2 Java statements 3.2.1 Expression Statements 3.2.2 Declaration Statements 3.2.3 Control-flow statements 3.3 Variables and Constants 3.3.1 Local Variables 3.3.2 Class Variables 3.3.3 Instance Variables 3.3.4 Integer constants 3.3.5 Real Constants 3.3.6 Single character constants 3.3.7 String constants 3.4 Java Data operations 3.4.1 Variable assignment 3.4.2 Variable reading 3.4.3 Variable arithmetic 3.4.4 Object Instantiation	• Practical tests • Oral tests • Written tests
	3.5 Java Program to perform an operation 3.5.1 Area of a circle 3.5.2 Solve Quadratic equations 3.5.3 Calculate compound interest	

4. Use Control structure	4.1 Java Control Statements 4.1.1 Decision making statements 4.1.2 Looping statements 4.1.3 Branching statements 4.2 Uses of different control statements in Java 4.2.1 Decision making statements 4.2.1.1 If then 4.2.1.2 If then else 4.2.1.3 Switch 4.2.2 Looping statements 4.2.2.1 for 4.2.2.2 while 4.2.2.3 do while 4.2.3 Branching statements 4.2.3.1 break 4.2.3.2 Continue 4.3 Creation of programs using control statements	• Practical tests • Oral tests • Written tests
5. Use Methods	5.1 Java Methods 5.1.1 Definition 5.1.2 Structure 5.2 Demonstration of methods 5.2.1 Creating Methods 5.2.2 Method calling 5.2.3 Void keyword 5.2.4 Passing parameters by value	• Practical tests • Oral tests • Written tests

	5.2.5 Method overloading 5.2.6 Using command line arguments 5.2.7 This keyword 5.2.8 Variable arguments 5.2.9 The finalize () method 5.3 Creation programs to implement methods	
6. Object Oriented Programming	6.1 Object oriented programming concepts 6.1.1 Inheritance 6.1.2 Encapsulation 6.1.3 Abstraction 6.1.4 Polymorphism 6.2 Classes 6.2.1 Declaring attributes 6.2.2 Creating Methods 6.3 Objects 6.3.1 Creating objects 6.3.2 Calling methods 6.4 Creation of programs to implement inheritance	• Practical tests • Oral tests • Written tests

Suggested Methods of Instruction

- Presentations and practical demonstrations by trainer;
- Guided learner activities and research to develop underpinning knowledge;
- Supervised practical assignments and projects;
- Visiting lecturer/expert from the ICT sector;
- Industrial visits.

Recommended Resources for 25 trainees

S/No.	Category	Item Description / Specifications	Quantity	Trainee:Item Ratio
1.	**Learning Materials**	Java Programming textbooks	25	1:1
2.		Java handouts, syntax reference sheets, sample programs	25 Sets	1:1
3.		Access to online platforms	25 Logins	1:1
4.	**Learning Facilities**	Computer lab/classroom with 25 workstations, whiteboard, projector	1 Room	Shared
5.	**Infrastructure**	Stable internet connection	1 Setup	Shared
6.	**Tools & Equipment**	Laptops/Desktops	25	1:1
7.		Java Development Kit (JDK) installed	25 Installs	1:1
8.		Eclipse IDE or IntelliJ IDEA Community Edition installed	25 Installs	1:1
9.		Code playgrounds (e.g., Repl.it, JDoodle, OnlineGDB for Java)	25 Logins	1:1
10.		Git & GitHub accounts for version control and code collaboration	25 Accounts	1:1
11.	**Consumable Materials**	Notebooks, pens, printed exercises, test papers	25 Sets	1:1
12.		USB drives or cloud storage access (Google Drive, OneDrive, etc.)	10	1:2.5
13.	**Safety & Support**	Antivirus software, surge protectors, fire extinguishers, basic first aid kit	5 Kits	1:5