Republic of Rwanda
Ministry of Education

RTB | RWANDA TVET BOARD

SPEOJ402

Fundamentals of OOP with Java

Apply Fundamentals of OOP with Java

Competence

RQF Level: 4

Learning Hours 110

Credits:11

Sector: ICT and Multimedia

Trade: Software Programming and Embedded Systems

Module Type: Specific Module

Curriculum: ICTSES4002 TVET Certificate IV in Software Programming and Embedded Systems

1200

Issue Date: August 2023

<table>
<tr><td>Purpose statement</td><td colspan="6">This module aims to equip leaners with knowledge and skills necessary to apply in software development using Java. By the end of this module, learners will be able to Describe Java basics, Implement Object Oriented Programming, Apply enums, regular expression and multithreading, Develop Graphical User Interfaces , Apply Java Network Programming, Apply Java Network Programming, Apply Java Database Connectivity.

The learners will be able to carry out the above tasks with confidence and minimum supervision.</td></tr>
<tr><td>Learning assumed to be in place</td><td colspan="6">Fundamentals of Programming with C, Basics of Databases Development, Discrete Mathematics</td></tr>
<tr><td rowspan="6">Delivery modality</td><td colspan="2">Training delivery</td><td>100 %</td><td colspan="2">Assessment</td><td>Total 100%</td></tr>
<tr><td colspan="2">Theoretical content</td><td>30%</td><td rowspan="4">Formative assessment</td><td>30%</td><td rowspan="4">50%</td></tr>
<tr><td colspan="2">Practical work:</td><td rowspan="3">70%</td><td rowspan="3">70%</td></tr>
<tr><td>Group project and presentation</td><td>30%</td></tr>
<tr><td>Individual project /Work</td><td>70%</td></tr>
<tr><td colspan="3"></td><td colspan="2">Summative Assessment</td><td>50%</td></tr>
</table>

Elements of Competence and Performance Criteria

Elements of competence	Performance criteria	
1. Describe Java basics	1.1.	Features of Java are properly described based on Java documentation
	1.2.	Java Development Environment is properly described based on Java language-specific features
	1.3.	Java classes and variables are properly introduced based on their types
2. Implement Object Oriented Programming	2.1.	OOP concepts are properly applied based on the Java programming standards
	2.2.	Input/output Streams are appropriately described based on the Java I/O API documentation
	2.3.	Exception Handling in Java is described in accordance with the Java documentation

	2.4.	Core Collection classes are appropriately defined based on the Collection API documentation
	2.5.	Generic classes are appropriately defined based on Java documentation
3. Apply enums, regular expressions and multithreading.	3.1.	The enumerated types are properly used according to the predefined constants.
	3.2.	The design patterns are efficiently implemented according to the software engineering principles.
	3.3.	The regular expressions are properly developed according to the java.util.regex package.
4. Apply Java Database Connectivity	4.1.	JDBC and JDBC Operations are accurately implemented based on the java.sql API documentation.
	4.2.	Blobs and Clobs are correctly handled according to the guidelines in the java.sql API documentation.
	4.3.	Transactions Management in JDBC is appropriately described based on the java.sql API documentation.
5. Develop Graphical User Interfaces	5.1.	JAVAFX Components are accurately developed based on Java programming Standards.
	5.2.	Layouts and Events are correctly designed according to the JAVAFX documentation.
	5.3.	Java Beans are appropriately described in line with the JavaBeans specifications.
6. Apply Java Network Programming	6.1.	Access Network Configuration is accurately described based on the networking principles.
	6.2.	JAVA Socket Programming is correctly described according to the Socket API in the java.net documentation.
	6.3.	JAVA Connection-oriented and connection-less protocol is appropriately described based on the Socket API in the java.net documentation.

Intended Knowledge, Skills and Attitude

Knowledge	Skills	Attitude
✓ Know Java fundamentals ✓ Comprehend OOP pillars and principles ✓ Grasp the concepts and objects, including constructors, fields, and methods ✓ Learn exception handling technics ✓ Understand I/O streams ✓ Handle file input/output operations ✓ Understand core collections and generic classes ✓ Understand enums, regular expressions ✓ Understand multithreading concepts ✓ Knowledge of Java concurrency utilities, synchronization, and thread safety ✓ Understand database operations	✓ Set up development environment ✓ Organize code ✓ Debug Java Code ✓ Handle exceptions ✓ Develop Java applications ✓ Perform unit testing ✓ Organize process execution ✓ Connect Java applications to databases ✓ Develop Graphical user interface ✓ Apply socket programming	✓ Have a problem-solving mindset ✓ Need for continuous learning ✓ Collaborate with others ✓ Encourage creative thinking ✓ Be open to feedback, code reviews, and new ways of doing things ✓ Use creativity and innovation throughout the software architecture ✓ Familiarize with testing tools ✓ Have positive attitude towards problem-solving and a willingness to tackle complex problems ✓ Be eager to learn about the latest advancements in Java and related technologies ✓ Have willingness to adapt to new tools, libraries, and best practices ✓ Be carefully to details, especially when it comes to writing clean, efficient, and maintainable code ✓ Work effectively in a team, communicate ideas clearly, and collaborate with team members

Course content

Learning outcomes **At the end of the module the learner will be able to:**

1. Describe JAVA basics
2. Apply Object Oriented Programming
3. Apply Enums, regular expressions and multithreading
4. Develop Graphical User Interfaces
5. Apply Java Network Programming
6. Apply Java Database Connectivity

Learning outcome 1: Describe Java Basics	Learning hours: 10

Indicative content

- **Description of java Features**
  - ✓ Install java environment
  - ✓ Platform Independent
  - ✓ Security
  - ✓ Portability
  - ✓ Object-Oriented
- **Description of Java Development Environment**
  - ✓ Description of JVM
  - ✓ Description of JRE
  - ✓ Description of JDK
  - ✓ Setting path and environment variables
- **Introduction to classes and variable**
  - ✓ Class definition
  - ✓ Types of classes
    - Static class
    - Final class
    - Abstract class
    - Concrete class
    - Singleton class
    - POJO class
    - Inner class
  - ✓ **Types of Java variables**
    - Local variables
    - Instance variables
    - Static variables
  - ✓ **Accessor and mutator methods**

✓ **Naming conventions**	
Equipment	Computer
Materials	• Internet • marker pen • Handout
Tools	Code editors, JDK, JRE
Facilitation techniques	• Brainstorming • Group discussions on Java Development Environment • Demonstrations on setting path and environment variables • Practical exercises on creating classes
Formative assessment methods /(CAT)	• Oral assessment • Written assessment • Practical assessment

Learning outcome 2: Implement Object Oriented Programming	**Learning hours: 40**

Indicative content

- **Description of Classes and Objects**
  - ✓ **Classes**
    - Class Members
    - Class Constructors (Default, Implicit, Explicit)
    - Use of 'this' and 'static' Keyword
    - Parameterized Constructors
    - Parameterized methods
    - Access modifiers
    - Getters and setters
    - Nested Classes
  - ✓ **Objects**
    - Introduction to Objects
    - Creation of Objects
    - Use of Objects
  - ✓ **JUnit**
  - ✓ **Memory Management**
    - Garbage Collection
    - Memory Allocation
    - The Stack and the Heap

- Static and Runtime Memory Allocation
- Memory Leaks and How to Avoid Them

Application of OOP concepts

✓ **Description of OOP pillars**

- Encapsulation
- Inheritance (Extends, Casting, Super)
- Polymorphism
- Abstraction (Abstract, Multiple inheritance, Interface)

✓ **Advanced Application of OOP concepts**

- Application of OOP Principles to create a Personalized Java Library ready to use

• Description of Input/output Streams

✓ **I/O Streams**

- Usage of Byte Streams to handle I/O of raw binary data.
- Utilization of Character Streams to handle I/O of character data.
- Usage of Buffered Streams to optimize input and output.
- Scanning and Formatting to allow a program to read and write formatted text.
- Usage of Data Streams to handle binary I/O of primitive data type and String values.
- Usage of Object Streams to handle binary I/O of objects.

✓ **File I/O**

- Description of the Path class and performing Path Operations
- Checking, moving, copying, deleting a File or Directory
- Managing file Metadata
- Creating, Writing and Reading Files
- Creating and Reading Directories
- Walking the File Tree
- Finding Files

• Exception Handling in Java

✓ **Exceptions overview**

- Definition
- Advantages of Exceptions

✓ **Catching and Handling Exceptions**

- The try Block
- The catch Blocks
- The finally Block
- The try-with-resources Statement

- ✓ **Defining and Throwing Exceptions**
  - Chained exceptions
  - Creating and throwing custom exceptions
- ✓ **Unchecked Exceptions**

Core Collection classes

- ✓ The Iterable Interface
- ✓ The Collection Interface
- ✓ The Set Interface
- ✓ Set Implementation Classes
- ✓ The List Interface
- ✓ List Implementation Classes
- ✓ The Queue Interface
- ✓ Queue Implementation Classes
- ✓ The Map Interface
- ✓ Map Implementation Classes

- **Generic classes**

- ✓ Generic classes and type parameters
- ✓ Implementing generic types
- ✓ Generic methods
- ✓ Constraining type parameters
- ✓ Type erasure

Resources required for the learning outcome

Equipment	Computer
Materials	• Internet • Didactic materials
Tools	Code editor software, JDK
Facilitation techniques	• Brainstorming • Group discussion on OOP concepts • Demonstration on the application of OOP concepts • Practical exercise on designing a Singleton, a personalized Java Library ready to use
Formative assessment methods /(CAT)	• Oral assessment • Written assessment • Practical assessment

Learning outcome 3: Apply Enums, design patterns, regular expression and multithreading.	**Learning hours:10**

Indicative content

- **Use of Enumerated Types**
  - ✓ Introduction to Enumerated types in Java
  - ✓ Significance of Enum in Java.
  - ✓ Use of Enum with predefined constants.
  - ✓ code readability and maintainability using Enums.
- **Implementation of Design Patterns**
  - ✓ Design patterns overview
  - ✓ Design patterns demonstration (Singleton, Factory, observer, Decorator, Adapter,..)
- **Development of Regular Expressions**
  - ✓ Introduction to regular expressions
  - ✓ Importance of regex in string manipulation.
  - ✓ Regular expression syntax
  - ✓ Dev of regex based on java.util.regex package.
    - Grouping
    - Capturing
    - Assertions
    - Matching (greedy & lazy)
    - Character escaping
  - ✓ Regex Performance considerations and best practices
- **Implementation of Multithreading**
  - ✓ Understanding the concept of multithreading
  - ✓ Benefits of multithreading in concurrent programming.
  - ✓ Concurrency vs Parallelism
  - ✓ Java's multithreading mechanisms:
    - Thread class
    - Runnable Interface
  - ✓ Java's Executor Framework.
    - Executors
    - ThreadPoolExecutor
    - ScheduledExecutorService
  - ✓ Multithreading best practices.
    - Thread Safety
    - Thread Communication
    - Avoiding Deadlocks
    - Concurrency Utilities
    - Multithreading on modern hardware

- ✓ Synchronization techniques
- ✓ Concurrency design patterns

Equipment	Computer
Materials	• Internet • Didactic materials
Tools	Code editors
Facilitation techniques	• Brainstorming • Group discussions on design pattern, regular expression and multithreading • Demonstrations on design pattern and regular expression • Practical exercises on design pattern and regular expression
Formative assessment methods /(CAT)	• Oral assessment • Written assessment • Practical assessment

Learning outcome 6: Apply Java Database Connectivity	**Learning hours: 20**

Indicative content

- **Implementation of JDBC and JDBC Operations.**
  - ✓ Basic concepts of JDBC
    - JDBC driver
    - JDBC Driver requirements
  - ✓ JDBC interfaces and classes
  - ✓ JDBC Adapter Database Operations
    - Insert
    - Update
    - Delete
    - SelectOne
    - SelectMultiple
    - SelectAll
  - ✓ JDBC Performance
    - Tuning
    - Metric
    - Counters
  - ✓ Transaction management in JDBC
    - Properties of transaction management
    - Advantages
    - Types of transactions

- **To handle Blobs and Clobs.**
  - ✓ Getting and Passing BLOB and CLOB Locators
  - ✓ Reading and Writing BLOB and CLOB Data
  - ✓ Creating and Populating a BLOB or CLOB Column
  - ✓ Accessing and Manipulating BLOB and CLOB Data
  - ✓ Additional BLOB and CLOB Features
- **Transactions Management in JDBC.**
  - ✓ Basic Transaction management concepts
  - ✓ Commit & Rollback
  - ✓ Using Savepoints

Resources required for the Learning outcome

Equipment	Computer
Materials	• Internet • Didactic materials
Tools	Code editors
Facilitation techniques	• Brainstorming • Group discussions on basic concepts of JDBC • Demonstrations on the implementation of JDBC Operations • Practical exercises on the implementation of JDBC Operations
Formative assessment methods /(CAT)	• Oral assessment • Written assessments • Practical assessments

Learning outcome 4: Develop Graphical User Interfaces	Learning hours: 20

Indicative content

- **JAVAFX Components are accurately developed based on programming Standards.**
  - ✓ **JavaFX Overview**
  - ✓ **JavaFX Architecture**
  - ✓ **JavaFX Application Structure**
  - ✓ **Creating the first JavaFX Application**
  - ✓ **JavaFX 2D Shapes**
    - Shape properties
    - JavaFX Line, Rectangle, Ellipse, Arc, Circle, Polygons
    - Cubic and Quad Curve
    - JavaFX Color and Gradient Color
  - ✓ **JavaFX UI**

- JavaFX UI Controls, Label, Button, RadioButton, CheckBox, TextField, PasswordField, Hyperlink, Slider, ProgressBar Progress Indicator, ScrollBar, FileChooser, Menu and Tooltip

✓ **JavaFX Text**

✓ **JavaFX Effects, Transformation, Animation**

✓ **JavaFX 3D Shapes**

✓ **JavaFX Charts**

✓ **JavaFX CSS**

✓ **Media with JavaFX**

Layouts and Events

✓ **JavaFX Layouts**

- JavaFX BorderPane
- JavaFX HBox
- JavaFX VBox
- JavaFX StackPane
- JavaFX GridPane
- JavaFX FlowPane

✓ **JavaFX Event Handling**

- JavaFX Convenience methods,
- JavaFX Event Filters,
- JavaFX Event Handlers

- **Java Beans are appropriately described in line with the JavaBeans specifications.**

✓ Java beans Basic Concepts
✓ Java beans Properties
✓ Accessing Java beans
✓ Accessing Java beans Properties

Equipment	Computer
Materials	• Internet • Didactic materials
Tools	Code editors, JDK, JRE
Facilitation techniques	• Brainstorming • Group discussions on Creating a JavaFX Application • Demonstrations on Creating a JavaFX Application Layouts design • Practical exercises on JavaFX Application development
Formative assessment methods /(CAT)	• Oral assessment • Written assessments • Practical assessments

Learning outcome 5: Apply Java Network Programming	**Learning hours: 10**

Indicative content

- **Describe Access Network Configuration**
  - ✓ Basic concepts of Network interface configurations
  - ✓ Retrieving Network interfaces with Java
  - ✓ Listing network interface addresses
  - ✓ Network interface parameters
- **Describe JAVA Socket Programming**
  - ✓ Socket Programming Concepts
  - ✓ Socket APIs and Libraries
  - ✓ Socket Interface Types
    - Stream sockets
    - Datagram
    - Raw socket
  - ✓ Socket Support in Network Protocols
- **Describe Connection-oriented and connection-less protocol**
  - ✓ Implementation of TCP and UDP Server Socket in Java
  - ✓ Implementation of TCP and UDP Client Socket in Java
  - ✓ Running the Java UDP Server and Client Socket
  - ✓ Monitoring UDP Network Traffic with Wireshark

Equipment	Computer
Materials	• Internet • Didactic materials
Tools	Code editors
Facilitation techniques	• Brainstorming • Group discussions on java Connection-oriented and connection-less protocol • Demonstrations on running Java UDP Server and Client Socket • Practical exercises on running Java UDP Server and Client Socket
Formative assessment methods /(CAT)	• Oral assessment • Written assessment • Practical assessment

Integrated/Summative assessment (For specific module)

Integrated situation

E-service is an eGovernment platform which enables the access and provision of government services in Rwanda, built within a PPP framework. Today E-service hosts many e-services deriving from different government agencies with more than 90,000 users a month. The services are available Online, On USSD and through a network of support agents. With the awareness building each day, E-service has also set out to harness the existing ecosystem (Telcos, Infrastructure, Human resources, payment gateways, etc...) by increasing its number of payment channels, access points & field agents, as well as its overall user rate. The service is designed to improve the citizen's way of life by making government services easier, faster and less costly to access. E-service plans to increase the amount of eServices accessible to citizens, and to increase the nationwide access points to get even closer to each citizen in the country.

To re-engineer government services so as to compliment the paperless and cashless economy that our country is striving towards. Expand from product to platform enabling an innovation ecosystem where Rwanda's youth can play a big part in the digital transformation of the country. The future is digital, and E-service plans to continue its journey on this path. IT manager was advised to hire back-end software developers to digitize new services using JAVA

You are assumed to be hired as a back-end developer to integrate E-service new systems features with other government agencies so that all information collected from those agencies can easily be put together for further processing and later to be accessed on the front-end for presentation to E-service users and/or to company's decision makers.

Before the intensive software development work, you are required to:

- In your project directory, create CSV files with dummy data simulating data from those government agencies.
- Develop java classes to access and read those CSV files.
- Store CSV files data into objects.
- Apply Object Oriented Programming concepts to do further development
- Persists those data in database by applying Java Database Connectivity
- Implement JDBC Operations
- Use exception handling where applicable

Timing: This task will take 5 hours

Resources

Tools	● Windows Operating Systems ● Internet ● Java ● Code Editors software

Equipment	Computer
Materials/ Consumables	● JRE (Java Runtime Environment) ● JDK (Java Development Kit)

Assessable outcomes	Assessment criteria (Based on performance criteria)	Indicator	Observation		Marks allocation
			Yes	No	
Learning outcome 1: Describe JAVA basics	1. 2Java Development Environment is properly described based on Java language-specific features	Ind 1.1. The environment is configured			5
		Ind 1.2. The tools used are explained			5
Learning outcome 2: Implement Object Oriented Programming	2.1 OOP concepts are properly applied based on the Java programming Standards.	Ind 1.1. The classes are used			10
		Ind 1.2 The naming conventions are implemented			10
		Ind 1.3 Encapsulation is used in the project			10
		Ind 1.4. Abstraction, inheritance is exploited			5
	2.2 Exception Handling in Java is described in accordance with the Java documentation.	2.1 Try and catch are used to handle Exceptions			5
		2.2 Throwable exceptions are handled			5
	2.3 Input/output Streams are appropriately described based on the Java I/O API documentation	Ind 1. Input/output Streams are used to access data			5

	2.4 Core Collection classes are appropriately defined based on the Collection API documentation	Ind 3.1 The collection classes are used.			
3.Apply enums, regular expressions and multithreading.	3.1 The design patterns are efficiently implemented according to the software engineering principles.	Ind 1.1 The design pattern used is explained			5
	3.2 The regular expressions are properly developed according to the java.util.regex package.	Ind 1.2 The regular expression is used			5
Learning outcome 6: Apply Java Database Connectivity	6.1 JDBC and JDBC Operations are accurately implemented based on the java.sql API documentation.	Ind. 1.1 JDBC Operations are implemented			10
		Ind.1.2 2Transaction management in JDBC			10
		1.3 Data in database are persisted			10
Total marks					
Percentage Weightage					100%
Minimum Passing line % (Aggregate): 70%					

References:

1. Herbert Schildt, **Java: The Complete Reference,** Twelfth Edition 12th Edition, McGraw Hill, 2021
2. Ian Darwin, **Java Cookbook: Problems and Solutions for Java Developers 4th Edition,** O'Reilly Media, 2020
3. Joshua Bloch, **Effective Java 3rd Edition,** Addison-Wesley Professional, 2017

4. Cay S.Horstmann, **Core Java for the Impatient 3rd Edition**, Addison-Wesley, 2023

Online resources

1. **https://docs.oracle.com/javase/tutorial/java/javaOO/arguments.html**
2. **https://www.javatpoint.com/collections-in-java**