| PRN. No. | | PAPER CODE | U114-3104 (Backlog |
|---|---|---|---|

DECEMBER 2024 (Backlog) EXAM Sem-I

# F.Y.(INFORMATION TECHNOLOGY)

(PATTERN 2023)

COURSE NAME: PROGRAMMING AND PROBLEM SOLVING-2

COURSE CODE: (IT12234)

Time: [2Hr]    [Max. Marks: 60]

(*) Instructions to candidates:

1) Use of scientific calculator is allowed
2) Use suitable data wherever required
3) All questions are compulsory. Solve any THREE sub questions from EACH question

| Que. No. | Question Description | Max. Marks | CO mapped | BT Level |
|---|---|---|---|---|
| Q1. | Solve any two sub questions from the following | | | |
| | A) Write and explain a simple Java program that takes input from the user using the Scanner class. Discuss how Java's data types, variables, and literals are utilized in this program, and compare the output mechanism in Java with that of C++. | [5] | I | 3 |
| | B) Analyze the use of control flow statements in Java by creating a program that involves if-else, switch-case, and for loop. How does Java's control flow differ from languages like C++? Justify your choice of control structures based on efficiency. | [5] | I | 4 |
| | C) Discuss the importance of packages in Java for organizing code. Evaluate how access modifiers (public, private, protected) work across different packages to protect data. Use examples to justify the impact of access control on software design. | [5] | I | 5 |
| | D) Apply the principles of encapsulation to design a class that protects sensitive data using private access modifiers and getter/setter methods. Evaluate the role of encapsulation in data security and class design. | [5] | I | 3 |
| Q2. | Solve any two sub questions from the following | | | |
| | A) Analyze the differences between inheritance and the IS-A relationship in Java. Provide an example where a class demonstrates inheritance and explain how the IS-A relationship is reflected in that example. What are the benefits and potential drawbacks of using inheritance in object-oriented design? | [5] | II | 4 |
| | B) Apply the concept of abstraction to design a Java system that separates interface from implementation. Use an abstract class and an interface to demonstrate different levels of abstraction. Discuss how these concepts contribute to a flexible design. | [5] | II | 3 |
| | C) Compare static binding and dynamic binding in Java with examples. Explain how method overloading and overriding are related to these types of binding, and evaluate which type of binding | [5] | II | 5 |

| | | | | |
|---|---|---|---|---|
| | is more efficient in different scenarios. | | | |
| | D) Differentiate between compile-time polymorphism and run-time polymorphism in Java, explaining how method overloading and method overriding demonstrate these types of polymorphism. Additionally, describe how static and dynamic binding are related to these concepts. | [5] | II | 3 |
| Q3. | Solve any two sub questions from the following | | | |
| | A) Explain the architecture and features of Java Foundation Classes (JFC), focusing on its role in building GUI applications. Compare the Swing library with AWT, highlighting how Swing improves flexibility and performance. | [5] | III | 4 |
| | B) Using the Swing API, design a simple Java application that includes various JComponent objects like JButton, JLabel, and JTextField. Explain the role of the JComponent class and analyze how it serves as the foundation for all Swing components. | [5] | III | 3 |
| | C) Design a Java Swing program that includes both JList and JComboBox components. Evaluate the use cases for lists and combo boxes in user interface design. Explain how to add items dynamically to both components at runtime. | [5] | III | 3 |
| | D) Demonstrate how the Action interface is implemented in a Java Swing application to manage button and menu actions. Analyze the advantages of using the Action interface over traditional event-handling mechanisms in Swing. | [5] | III | 4 |
| Q4. | Solve any two sub questions from the following | | | |
| | A) Explain the different types of JDBC drivers (Type 1 to Type 4) with examples of their usage. Compare and contrast their performance, portability, and suitability for different database environments. | [5] | IV | 4 |
| | B) Apply the basic steps of JDBC to establish a connection with a MySQL database. Explain the role of DriverManager, Connection, and Statement objects in this process, and evaluate the importance of closing the connection to manage resources effectively. | [5] | IV | 3 |
| | C) Compare and contrast the usage of Statement, PreparedStatement, and CallableStatement in JDBC. Analyze their advantages in terms of performance, security (SQL Injection), and flexibility, using relevant examples. | [5] | IV | 4 |
| | D) Explain the concept of Database Metadata in JDBC. Write a Java program that retrieves and displays metadata about a connected database, such as the number of tables, their names, and supported SQL features. Evaluate how database metadata can help in database-driven application development. | [5] | IV | 5 |

**BT LEVELS: 1:** Remember, **2:** Understand, **3:** Apply, **4:** Analyze, **5:** Evaluate, **6:** Create