VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

Reg. No

Final Assessment Test(FAT) - Apr/May 2025

Programme	M.C.A.	Semester	Winter Semester 2024-25
Course Code	PMCA502L	Faculty Name	Prof. Jayasudha M
Course Title	Java Programming	Slot	A1+TA1
		Class Nbr	CH2024250501721
Time	3 hours	Max. Marks	100

Instructions To Candidates

- Write only your registration number in the designated box on the question paper. Writing anything elsewhere on the question paper will be considered a violation.

Course Outcomes

CO1: Ability to familiarize with core object oriented concepts in Java

CO2: Apply multithreading and exception handling concepts in Java to solve problems

CO3: Design and develop GUI applications that involve databases for real world problems using JavaFX and JDBC

CO4: Design, develop and deploy web applications using servlets and Java server pages

CO5: Ability to write client server applications using networking concepts and cofigure spring applications with spring framework and hibernate

Answer all Questions (10 × 10 Marks)

01. Design a Java program to manage employee records using an 'Employee' class. This class should have attributes like 'name', 'employeeID', and an array 'salaries' to store monthly salaries for different months. Implement a constructor in the 'Employee' class using the 'this' keyword to initialize these attributes. Additionally, include methods to display employee details, calculate the average salary, and update the salary for a specific month. Utilize the 'Scanner' class for user input and ensure error handling for invalid input, such as non-numeric or negative salaries. Create an 'EmployeeManager' class with a main method to demonstrate the functionalities of the Employee class. The program should allow users to create multiple Employee objects, update their salaries, and display their details.

[10] (CO1/K1)

02. Design a Java program for a library management system that utilizes inheritance, method overloading, and method overriding. Implement a superclass 'Item' with attributes like 'title', 'itemID', and 'availableCopies'. Include methods for 'checkout', 'return', and displaying 'item information'. Extend this class to create subclasses 'Book' and 'DVD', representing different types of library items. In the 'Book' subclass, override the 'checkout' method to include an additional parameter for the due date. Override the 'checkout' method in the 'DVD' subclass to implement specific restriction checks, such as age restrictions for certain DVDs. Implement input validation to handle errors such as trying to check out an unavailable item, incorrect due date format, and invalid item details.

[10] (CO1/K1)

03. Develop a **Java program** that demonstrates various exception-handling scenarios within **one centralized class** (2 M). The program should:
a. Define a **custom exception** named InvalidInputException to handle situations where the user provides incorrect input for a specific operation. (2 M) b. Perform mathematical operations (e.g., addition and division) with error handling for ArithmeticException. (2 M) c. Simulate a scenario involving a Customer object where accessing attributes without initialization triggers a NullPointerException. (2 M) d. Include a simulated method interacting with a database, handling SQLException. (1 M) e. Include a method that manipulates an ArrayList, with handling for IndexOutOfBoundsException. (1 M)

[10] (CO2/K1)

04. Design a JavaFX program for a basic 'temperature converter' application. The application should feature a graphical user interface (GUI) with input fields for entering temperature in Celsius or Fahrenheit, and buttons to convert between the two units. Write code to perform temperature conversion calculations when the corresponding conversion button is clicked. Write error handling to manage scenarios such as invalid input values. Include functionality to clear the input fields and reset the converter to its initial state.

[10] (CO2/K1)

05. Develop a JavaServer Pages (JSP) application for an online movie database. Create a homepage (index.jsp) that welcomes users to the movie database and presents a list of available movies with details such as title, director, and release year. Implement a search form allowing users to search for movies by title, director, or genre, with dynamic handling to display matching results.

[10] (CO3/K1)

06. Describe the Model-View-Controller (MVC) architectural pattern in detail, delineating the responsibilities of each component (Model, View, and Controller) and elucidating how they collaborate within the system. Evaluate the significance of MVC in software engineering, emphasizing its advantages. Provide concrete examples of well-known frameworks or platforms that leverage MVC architecture, illustrating its practical implementation and effectiveness in real-world scenarios.

[10] (CO4/K1)

07. Explain the purpose and functionality of Hibernate in Java-based applications. Discuss how Hibernate simplifies database interactions by providing an object-relational mapping (ORM) framework. Describe the key features of Hibernate, such as automatic table creation, CRUD operations, and transaction management. Provide an example illustrating the usage of Hibernate to persist and retrieve data from a relational database.

[10] (CO4/K1)

08. Develop a Java web application using a Servlet and JDBC to manage a simple "Student Registration System." Create a Servlet named StudentServlet that handles HTTP GET and POST requests. The application should:
a. Display a list of all registered students (name, roll number, and course) from a database table students when a GET request is made. (2 M)
b. Allow users to submit a new student record (name, roll number, course) via a POST request, inserting it into the students table. (2 M)
c. Use JDBC to connect to a MySQL database named student_db, with the table students having columns: name (VARCHAR), roll_no (VARCHAR, primary key), and course (VARCHAR). (2 M)
d. Implement error handling for database connection failures and duplicate roll numbers (since roll_no is a primary key). (2 M)
e. Forward the response to a JSP page (students.jsp) to display the list of students or an error message if something goes wrong. (2 M)
Provide the complete code for StudentServlet, including JDBC setup, and a brief explanation of how the Servlet interacts with the JSP page and database.

[10] (CO5/K1)

09. Imagine you are developing a distributed banking system using Java RMI. The system consists of a server that manages customer accounts and a client that allows users to perform operations like checking their balance, depositing money, and withdrawing money. The server registers a remote object with the RMI Registry, and the client looks up this object to interact with the server. Your task is to
a. Define the remote interface for the banking system. (3M)
b. Implement the server-side logic and register it with the RMI Registry. (3.5 M)
c. Write a client that connects to the RMI Registry and performs a sample transaction. (3.5 M)

[10] (CO5/K1)

10. Develop a **file transfer system** using socket programming in Java. The system consists of a server that listens for incoming client connections and saves files sent by clients to its local filesystem and a client that uploads a file to the server. Your task is to
a. Implement the server, which listens on a specific port (e.g., 6000) and saves the received file with a predefined name (e.g., "uploaded_file.txt"). (4 M)
b. Implement the client, which connects to the server and sends a file from the local filesystem (e.g., "sample.txt"). (4 M)
c. Ensure the system handles file transfers reliably and closes connections properly. (2 M)

[10] (CO5/K1)

BL-Bloom's Taxonomy Levels - (K1-Remembering,K2-Understanding,K3-Applying,K4-Analysing,K5-Evaluating,K6-Creating)