MITE
Invent Solutions

MANGALORE INSTITUTE OF TECHNOLOGY & ENGINEERING

(A Unit of Rajalaxmi Education Trust®, Mangalore)
Autonomous Institute affiliated to VTU, Belagavi, Approved by AICTE, New Delhi
Accredited by NAAC with A+ Grade & ISO 9001:2015 Certified Institution

Model Question Paper

Third Semester MCA Degree Examination

Advanced Java

Time: 3 Hours **Max. Marks: 100**

Note: 1. Answer any FIVE full questions, choosing ONE full question from each module.
2. M: Marks, L: RBT (Revised Bloom's Taxonomy) level, C: Course outcomes.

		Module -1	M	L	C
Q1	a.	Outline the interface layout of an online quiz application using suitable AWT components. Demonstrate the components you would choose for displaying questions, presenting multiple options, and enabling navigation and submission.	10	L3	CO1
	b.	Create a GUI with three buttons labeled "Red", "Green", and "Blue". When a button is clicked, the background color of the window should change accordingly. Use event listeners to handle each action and ensure responsiveness.	10	L3	CO1
		OR			
Q2	a.	Implement a Java Swing-based GUI application for a simple calculator that performs addition of two numbers using a button click event. Demonstrate appropriate event handling and illustrate how your design ensures a user-friendly interface and modular code structure.	10	L3	CO1
	b.	Build a Swing-based interface combining multiple components (JList, JCheckBox, JTextArea, JButton) and containers. How would you manage user interactions through event handling?	10	L3	CO1
		Module- 2			
Q3	a.	Design a Java Swing application integrated with a backend database using JDBC. Explain the complete JDBC connectivity process, including establishing the connection, executing queries, and processing results. Justify your design with respect to code modularity and efficient database interaction.	10	L3	CO2
	b.	An educational institution maintains student information in a MySQL database, where a stored procedure accepts a student ID and returns the corresponding student name. Write a Java program using **CallableStatement** in JDBC to invoke this stored procedure, and explain the steps involved in passing input parameters and retrieving output parameters.	10	L3	CO2
		OR			
Q4	a.	Build a Java–MySQL JDBC application to insert and retrieve data using **PreparedStatement**, and justify how it improves security and performance over Statement.	10	L3	CO2

	b.	A software development team is building a Java application that must work with different relational databases without changing the source code. To achieve this, the application needs to dynamically obtain information about the connected database, such as its name, supported features, available tables, and column details. Explain how DatabaseMetaData in JDBC can be used to retrieve such information?	10	L3	CO2
Module – 3					
Q5	a.	A Java application needs to store and process an ordered collection of elements where duplicate values are allowed and frequent insertion, deletion, and access operations are required. In this context, explain the **List** collection interface in Java, describe its key characteristics and commonly used implementations,	10	L3	CO3
	b.	Develop a Java program to manage user-defined objects using different Collection interfaces—List**,** Set, and Queue. Demonstrate how each interface handles data insertion**,** duplicates**,** and element order. For each collection type, explain why it is suitable for specific operations and provide examples illustrating its behavior.	10	L3	CO3
OR					
Q6	a.	A Java application needs to store a collection of unique elements, where the order of storage may or may not matter depending on the requirement. In this context, explain the **Set** interface in the Java Collection framework, describe its key characteristics and commonly used implementations, and analyze how selecting different `Set` types impacts element storage, retrieval, and overall performance.	10	L3	CO3
	b.	Create a Java program to simulate a generic storage system that can store different types of items using generic classes and interfaces. Incorporate wildcard arguments and demonstrate how type constraints and type inferences are applied. Analyze how the use of generics reduces code duplication and improves type safety in real-time applications.	10	L3	CO3
Module – 4					
Q7	a.	Implement a CRUD application using Spring Boot and Spring Data JPA. Choose how to map entities to a database and analyze how repository interfaces reduce boilerplate code. Justify the advantages of JPA in data handling.	10	L3	CO4
	b.	Write a Spring Boot application that connects to a relational database using JPA. Include entity creation, data persistence, and query operations. Make use of Spring Boot auto-configuration and dependency injection to reduce manual configurations	10	L3	CO4
OR					
Q8	a.	Build a reactive web application using Spring WebFlux. Compare the programming model of reactive applications with traditional Spring MVC. Identify the benefits of reactive programming in terms of performance and scalability	10	L3	CO4
	b.	Develop a RESTful web service using Spring Boot to manage a list of user-defined entities Explain the annotations and components used to build the REST API and justify how Spring Boot simplifies RESTful application development.	10	L3	CO4
Module – 5					
Q9	a.	With the help of sample application, Illustrate the lifecycle of a Hibernate entity using a sample application. Identify how these states influence data operations and session management in Hibernate-based systems.	10	L3	CO4
	b.	Develop a Hibernate-based application that demonstrates advanced query operations such as group by, join using HQL. Identify their role in optimizing data retrieval.	10	L3	CO4
OR					

Q10	a.	Apply the advanced entity associations such as one-to-one (bidirectional), using join tables, and inheritance mapping in a sample Hibernate application. How do these mappings support real-world business modeling and affect database normalization.	10	L3	CO4
	b.	Create and execute queries using Hibernate Query Language (HQL) to retrieve the data from a mapped entity. What do you think are the advantages of HQL over native SQL in terms of object-oriented data access.	10	L3	CO4