

JK Lakshmipat University Jaipur
INSTITUTE OF ENGINEERING AND TECHNOLOGY
 End Term Examination (Regular/Backlog/Additional), December 2025
 B.Tech Semester V, 2025-26

Roll No.....

CS1101: Object Oriented Programming

Time: 2.5 hours

Max. Marks: 30

Instructions to students:

1. Do not write anything other than your roll number on question paper.
2. Mention all the assumptions for your answers clearly.
3. Write the answers to the point.

1	(a) Why would a developer choose an abstract class instead of an interface in Java? (b) WAP in java following the instruction Define an abstract class named Vehicle containing: • An abstract method move() that must be implemented by subclasses to describe how the vehicle moves. • A constructor or message to indicate when a Vehicle is created. Intermediate Class – LandVehicle, Inherit from the Vehicle class: • Provide a concrete implementation of the move() method that prints: "The vehicle moves on land." • Include a constructor to indicate creation (for clarity in multilevel inheritance flow). Derived Class – Car, Inherit from the LandVehicle class • Add a new method fuelType() which prints: "Fuel type: Petrol". In your main() method or equivalent: • Create an object of the Car class. • Call both move() and fuelType() methods using the Car object. • Show output that demonstrates the correct inheritance and behavior.	(CO1, CO2) 1.0 + 4.0
2.	Guess and elaborate the following concepts with providing suitable examples: (a) Java's mechanism for dealing with runtime anomalies, elaborating on each keyword (b) Java's object-creation initialization process. (c) Java's mechanism that allows a class to derive attributes and methods from another class (d) Java's keyword used to prevent modification of variables, methods, and classes	(CO4) 2.0*4= 8
3.	Your task is to define an interface named Payment, and implement it in two classes: CreditCardPayment and UPIPayment. 1. Interface – Payment: Define an interface named Payment with the following method: void makePayment(double amount); (This method will be implemented differently depending on the payment type.) 2. Class – CreditCardPayment Implements the Payment interface. • Additional attribute: cardNumber (String) • Constructor to initialize the cardNumber.	(CO2, CO4) 4.0

	- Implement the makePayment() method to print: - Payment of \$<amount> made using Credit Card ending with <last 4 digits of card number>. 3. Class – UPIPayment Implements the Payment interface. - Additional attribute: upiId (String) - Constructor to initialize the upiId. - Implement the makePayment() method to print: Payment of \$<amount> made using UPI ID <upiId>. 4. Main Method in your Main class: - Create one object of each payment type: - CreditCardPayment with a sample card number - UPIPayment with a sample UPI ID - Call makePayment() on each object with a sample amount.	
4.	(a) Explain the role and applications of the super keyword in Java programming, with examples. (b) You are developing an inventory management system for an electronics store. The system should represent different categories of products using multilevel inheritance. You must implement three classes: Product (base class), Electronic (intermediate class), and Smartphone (derived class). Each class should demonstrate the use of constructor chaining using the super keyword and method overriding. 1. Base Class – Product Create a class named Product with the following: - Attributes: - productName (String): Name of the product - price (double): Price of the product - Constructor: - Takes productName and price as parameters and initializes the attributes. - Method: - void display() that prints: Product Name: <productName> and Price: \$<price> 2. Intermediate Class – Electronic Create a class Electronic that inherits from Product and adds more features: - Additional Attribute: - warrantyPeriod (int): Number of months the product is under warranty - Constructor: - Takes productName, price, and warrantyPeriod as parameters - Uses super() to call the Product constructor - Method: - Override display() method - First call super.display() to print basic product details - Then print: Warranty Period: <warrantyPeriod> months 3. Derived Class – Smartphone Create a class Smartphone that inherits from Electronic:	(CO1, CO5) 2.0 + 4.0

	• Additional Attribute: ◦ operatingSystem (String): OS used in the smartphone (e.g., Android, iOS) • Constructor: ◦ Takes productName, price, warrantyPeriod, and operatingSystem ◦ Uses super() to call the constructor of the Electronic class • Method: ◦ Override display() method ◦ First call super.display() ◦ Then print: Operating System: <operatingSystem> 4. Main method in your Main class: • Create an object of the Smartphone class with appropriate sample values • Call the display() method on the object	
5.	(a) Write a program in java to remove substring from sentence and print the sentence without it. (b) Write a Java program that does the following: 1. Create a method validateScore(int score) that: • Throws a custom checked exception called InvalidScoreException using the throw keyword if the score is less than 0 or greater than 100. • Declare this exception using the throws keyword. 2. In the main method: • Call validateScore() inside a try block with a test score. • Catch the InvalidScoreException using a catch block and print an error message. • Include a finally block that prints "Validation complete" regardless of whether an exception occurred. 3. Define the InvalidScoreException class as a checked exception (i.e., extend Exception).	(CO3) 4.0 (CO2) 3.0

