VIT-AP UNIVERSITY	Final Assessment Test –Winter (2024-25)–MAY 2025	
	Maximum Marks: 100	Duration: 3 Hours
Course Code: CSE2005	Course Title: Object Oriented Programming using JAVA	
Set No: 4	Exam Type :~~Open Book/Open Notebook~~/Closed Book	School: SCOPE
Date: 24/05/2025	Slot: C2	Session: FN

Keeping mobile phone/smart watch, even in 'off' position is treated as exam malpractice

General Instructions if any:

1. **"*fx* series" - non Programmable calculator are permitted : NO**
2. **Reference tables permitted : YES / NO (if Yes, Please specify): NO**
3. **Assume data whenever necessary and clearly state the assumptions.**

PART – A: Answer any <u>TEN</u> Questions, Each Question Carries 10 Marks (10×10=100 Marks)

1. An array to store the prices of all the items in the shopping cart. Use a loop to go through each item one by one. For each item check if the item's price is over $50. If it is, apply a 15% discount by multiplying the price by 0.85 (since 100% - 15% = 85%, and 85% = 0.85 in decimal). If it's not over $50, we leave the price as it is. Keep a running total by adding up each item's (possibly discounted) price using the + operator. At the end, the total will represent the final amount the customer needs to pay. **(10 M)**

   Test Case

   Items: [30, 60, 20, 80, 45]

   $30 → no discount → $30

   $60 → 15% discount → $60 × 0.85 = $51

   $20 → no discount → $20

   $80 → 15% discount → $80 × 0.85 = $68

   $45 → no discount → $45

2. In Java Programming a constructor Declare timestamp, userId, and actionType fields as private final. Initialize these fields through the constructor only. Capture timestamp using System.currentTimeMillis() or LocalDateTime.now(). Accept userId and actionType as constructor parameters. Do not create any setter methods to modify these fields. Provide only getter methods to access the values safely. **(10M)**

   Sample Output

   Timestamp: 2025-04-28T14:32:10.523

   User ID: user123

   Action Type: LOGIN_ATTEMPT

3. A singing competition is being organized, and a list of candidates needs to be managed. Create an abstract class **SingingCompetition** with an abstract method **ReadDetails()** and a method DisplayVenue() to display the completion venue details. Create a class named Singers which is a subclass of **SingingCompetition** with the following attributes: **name:** Name of the singer **age:** Age of the singer **living_city:** City where the singer lives **singer_id:** A unique ID assigned to each singer, a method **Search()** that allows searching for a singer by singer_id and a method **Count()** to count how many singers are above 30 years of age. Create an array of n objects of the Singers class in the main function and display the details **(10M)**
4. Create a Java program that implements the multiple inheritance.

• Create an interface **Discount** with abstract method calculate_Discount ().

• Create a Class **Books**, with data members book name, author, booktype (type can be "Paper"or " kindle") and price.

•Create a class, **Customers** which inherits Books and implements Discount, the method calculate_Discount () calculate the discount depending on the book type if it is paper, discount is 15% and if it is kindle discount is 10 % based on booktype chosen by customer. Create object for 3 customers and display the details. **(10M)**

5. Create a class named BookStall with the attribute String title, int ID, int qty and double price which are private. Within the class, include a constructor that requires the book title, Include an method named setPrice(int qty, double price). Create two child classes of BookStall: Technical and NonTechnical. Each must include a setPrice(int qty, double Price) method that sets the price. Write a constructor for each subclass, and include a call to setPrice() within each. Write a class demonstrating that you can create both a Technical and NonTechnical Book, and display their fields with total cost (qty*Price). **(10M)**
6. Write a Java program to create a class Exam class that models a test scenario with the private attributes duration and status fields (scheduled or not scheduled). Timer is a non-static inner class, it can directly access private members of its enclosing class (Exam) without needing getters or setters. It has the details of student name, regno and a constructor to initialize and calls the super class constructor, a method to display all the details. Write a main method to store details of 5 students and display it. **(10M)**
7. In a Weather Monitoring System, different sensors details such as Temperature and Pressure. Create Concrete Class for Each Sensor Type: TemperatureSensor class has a method getTemp() and an attribute temp to read the temperature value if the temperature read is >40 throw an ArithmeticException, PressureSensor class has a method getPressure() and an attribute pressure to read pressure and if pressure is >100 throw an NullPointerException. Create a class SensorData that inherits TemperatureSensor and Pressure Sensor class and has a method display() to display temperature and Pressure. Create an object and call the method getTemp() and getPressure() to display the details. Assume the attributes for each sensor type. **(10M)**
8. Fitness Tracker Feature of a sports application, players log their physical activity durations (e.g., running, gym, swimming) in a daily log. If a player logs more than 24 hours of total activity in a day, this is logically invalid, since a day only has 24 hours. We create a custom exception: ActivityLimitExceededException, which is thrown when the total exceeds 24 hours. Additionally, activity data might come from a external source in String format, like "2.5", "1.0", etc. If the data is incorrectly formatted (e.g., "two hours"), it causes a ParseException, which we catch and handle. Write java program for the above statement. **(10M)**

Sample Input:

String[] logs = {"2.5", "5", "six", "10", "8"};

sample output

Total activity logged: 2.5 hrs.
Total activity logged: 7.5 hrs.
Data format error: 'six' is not a valid number.
Total activity logged: 17.5 hrs.
Exception: Logged activity exceeds 24 hours: 25.5 hrs.

9. Create a class Hotel order with attributes tiffen tiffenprice, tiffenqty, lunch, lunchprice and lunchqty and totalprice. Create a constructor to initialize all the parameters and calculate the total price in the constructor. Create a HashMap by passing Integer and Hotel as type to store the orderid and the

Hotel object. Create 5 objects in the main method and store the values in the HashMap. Also, print the HashMap with the details of order and totalprice for all the 5.Write a java program to implement the above. **(10 M)**

10. Design a generic stack class called GenericStack<T> in Java that can store and manage different types of objects. The class should be able to work with any Wrapper classes (such as Integer, Double) without requiring type casting when elements are pushed or popped from the stack.

    Generic Type Declaration: Ensure the class uses a type parameter T to represent the type of object the stack will hold.

    Push Method: Implement the push(T item) method that adds an item to the stack.

    Pop Method: Implement the pop() method that removes and returns the top item from the stack. Make sure the method returns the correct type without casting.

    Storage for Stack: Internally, use an array or a linked list to store the elements of the stack. Handle dynamic resizing if necessary.

    Size Method: Implement a method to return the current size of the stack. **(10M)**

11. Create a class Employee details with attributes name, id, salary and dept and a constructor to initialize the values. Write two methods checkname(String name) to check whether the name has a character 'A' and print "valid name" otherwise "not a valid name" and a display() method to display all the details of the Employee. Create two thread, where the first thread checks the name and second thread print the details. **(10M)**

12. VIT University, faculty members can apply for summer vacation in 3 different slots, each lasting 4 weeks. However, to maintain seamless operations and coverage, no more than ten faculty members may select the same vacation slot. Create a class Vacation with attributes char slottype (values can be slot1-A, slot2-B, slot3-C) and slot1, slot2 and slot3 all initialized to 2. Write a method slotBooking() to book the slot and display allotted slot with name and id otherwise "Not Available Choose a different slot". The slot booking procedure is mechanised by Java multithreading, with each faculty member represented as a separate thread. When accessing or modifying shared slot data, synchronization is essential to ensure thread safety and prevent inconsistent slot booking. Create 3 Thread class with the attributes facultyname and id and a constructor to initialize. Also, call slot booking to display the selected slot. **(10M)**

QP MAPPING

Q. No.	E/A/T	Module Number	Marks	BL	CO Mapped	PO Mapped	PEO Mapped	PSO Mapped
Q1	A	1	10	3	CO1	PO1	PEO4	
Q2	E	1	10	2	CO1	PO1	PEO4	
Q3	A	2	10	3	CO2	PO1	PEO4	
Q4	T	2	10	3	CO2	PO1	PEO4	
Q5	A	3	10	3	CO3	PO2,11	PEO2	
Q6	E	3	10	3	CO3	PO2,11	PEO2	
Q7	A	4	10	4	CO4	PO2,5	PEO2	
Q8	E	4	10	2	CO4	PO2,5	PEO2	
Q9	A	5	10	3	CO5	PO3	PEO3	
Q10	A	5	10	3	CO5	PO3	PEO3	
Q11	T	6	10	4	CO6	PO2,3,5	PEO3	
Q12	T	6	10	3	CO6	PO2,3,5	PEO3	