SVKM'S NMIMS

MUKESH PATEL SCHOOL OF TECHNOLOGY MANAGEMENT& ENGINEERING/

SCHOOL OF TECHNOLOGY MANAGEMENT

Academic Year: 2023-2024

Program/s: **B. Tech, MBA Tech and BTI**   Year: **II/III/IV** Semester: **IV/VI/VIII**

Stream/s : **Computer Engineering, Computer Science and computer**

Subject: **Object Oriented Programming Through JAVA**   Time: **2 hrs.** (10:00 to 12:00)
                                                                          an   noon

Date: **29 / 4 / 2024**   No. of Pages: **3**

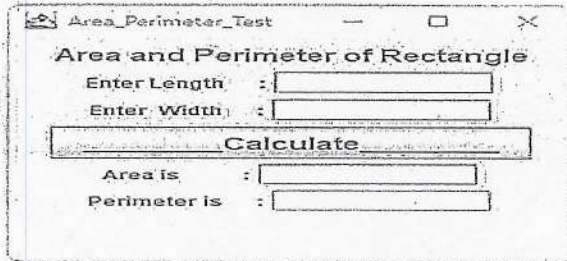Marks: **50**

Final Examination/Re-Examination (2022-23)

Instructions: Candidates should read carefully the instructions printed on the question paper.

1) Question No. **1** is compulsory.
2) Out of remaining questions, attempt any **3** questions.
3) **In all ___4___ questions to be attempted.**
4) All questions carry equal marks except Question 1
5) **Figures in brackets on the right hand side indicate full marks.**
6) **Assume Suitable data if necessary.**

Q1		Answer the following questions all are compulsory	[20]
CO-1 ; SO-1 ; BL-1	a.	Illustrate a method named "enough" takes one integer argument "goal" and returns the smallest positive integer n for which the sum of consecutive positive integers from $1\ to\ n$ is at least equal to "goal". Implement in JAVA using class and method. For instance: enough(9) that returns 4 because $1 + 2 + 3 + 4 \geq 9$	[5]
CO2- ; SO-6 ; BL-2	b.	Illustrate a class named Eligible that has data members as number of classes held, and attended, as well as methods: input () and compute (). The input() accepts the data from user and compute() calculates the percentage and verifies that a student having 80% or more then displays a message as " student permitted to appear in the exam" otherwise "student not allowed to appear in the exam". Create an instance of class that invokes the both the methods.	[5]
CO- 4; SO-1 ; BL-4	c.	Create a user form using swings that contains two buttons, namely: "SUBMIT" and "CANCEL" • On clicking "SUBMIT" button, "submitted" should get displayed • On clicking "CANCEL" button, "cancelled" should get displayed	[5]

CO-3 ; SO-2 ; BL-3	d.	Apply try-catch blocks in java to demonstrate exception handling for ArrayIndexOutOfBoundsException, as well as print error messages.	[5]	
Q2 CO-2; SO-7; BL-5		An electricity board charges the following rates to domestic users to discourage large consumption of energy : For the first 100 units – 50 P per unit Beyond 300 units – 60 P per unit If the total cost is more than Rs.250.00 then an additional surcharge of 15% is added on the difference. Define a class Electricity in which the method Bill computes the cost. Define a derived class More_Electricity, and override Bill to add the surcharge. Develop the java code by using the constructor, super keyword and method override concepts.	[10]	
Q3 CO-2; SO-7; BL-6		Create a class named 'Member' having the following members: Data members : Name, Age, Phone number, Address, Salary. It also has a method named 'printSalary()' which prints the salary of the member. Two classes 'Employee' and 'Manager' inherits the 'Member' class. The 'Employee' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an employee and a manager by making an object of both of these classes and print the same.	[10]	
Q4 CO-3; SO-7; BL-3		Demonstrate the hybrid inheritance for the following scenario Create interfaces LandAnimal and AquaticAnimal with move() and swim() methods respectively. The Animal class defines makesound() that prints the message as" Animal is making a sound". LandMammal class inherits Animal and LandAnimal, AquaticMammal class inherits Animal and AquaticAnimal, and HybridMammal class inherits LandMammal and AquaticAnimal by implementing relevant methods. Create the objects for LandMammal, AquaticMammal and HybridMammal and call the respective methods.	[10]	
Q5 CO-4; SO-1;		Develop Java program that takes an ArrayList of integers as input and returns the sum of all the elements in the list.	[10]	

BL-6			
Q6 CO-4; SO-1; BL-3		Demonstrate a Java application featuring an event-driven graphical user interface, as depicted below, to calculate the area and perimeter of a rectangle. Your program ought to utilize text fields to input the length and width, and upon clicking the "__Calculate__" button, display the computed area and perimeter. 	[10]