MANIPAL INSTITUTE OF TECHNOLOGY

MANIPAL

(A constituent unit of MAHE, Manipal)

II SEMESTER MCA IN-SEMESTER EXAMINATION JUNE 2021

SUBJECT: Java Programming [MCA 4253]

Date of Exam: **09/06 /2021** Max. Marks: **30**

Time of Exam: **10:30 AM – 12:45 PM (135 minutes: 120 + 15 for submission)**

Note: Answer all the questions.

1A.	What is the output of the following code? Justify the reason for the same and possible solution if necessary to make the program work.
	public class A
	{
	public static void main(String[] argv)
	{
	char c = 'c';
	int n = 88;
	c = n;
	System.out.println(" c = "+c);
	}
	}
		2

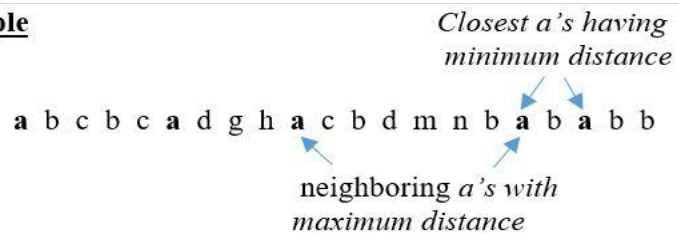
1B.	Given a method, void compute_() { } Can it be prevented from being overridden? If so, write the relevant statements to illustrate the same.	**2**
1C.	Given two classes, namely, **Student (**RegNo**,** Name , address details , AvgMarks **)** and **Address (** City, District ,Pincode**).** Give a suitable class structure to represent the relationship between these two classes using the object-oriented features of Java.	**2**

1D.	Create a class called *Employee* (instance fields: empID, name, age, Salary) with necessary constructor/methods corresponding to the Employee instances used in the EmployeeClassDemo:
	`public class EmployeeClassDemo`
	`{`
	`public static void main(String[] args)`
	`{`
	`Employee[] staff = { new Employee( 1, "Anil", 25, 50000 ),`
	`new Employee( 2, "John", 35, 60000 ),`
	`new Employee( 3, "Vinod", 38, 40000 )`
	`};`
	`for( Employee e : staff )`
	`e.raiseSalary( 5 ); // Raise everyone's salary by 5%`
	`for ( Employee e : staff )`
	`System.out.println( e );`
	`}`
	`}`
		2

	OUTPUT:
	`Emp id: 1`
	`Name: Anil`
	`Age : 25`
	`salary : 52500.0`
	
	`Emp id: 2`
	`Name: John`
	`Age : 35`
	`salary : 63000.0`
	
	`Emp id: 3`
	`Name: Vinod`
	`Age : 38`
	`salary : 42000.0`	

2A.	Assume that **method_2 ()** is called within **method_1 () ,** and **method_1 ()** is called in **main ().** It is given that **method_2** throws two exceptions: **MyException1** and **MyException2.** Illustrate any two ways to handle these exceptions with an example for each.	**3**
2B.	Write a java program to input Registration number, city, average mark, email id of the student, and validate the input data as per following criteria using exception handling technique: **i)** Average mark must be between 0 and 100. **ii)** Check for validity of email id (valid if ends with gmail.com or manipal.edu)	**3**
2C.	Develop a multithreaded application in java to perform array operations as follows: **MainThread:** Read input elements from the keyboard and store them in an array, also display the count of odd and even numbers obtained from the following threads: **Thread-1:** Display all the odd numbers. **Thread-2:** Display all even numbers. Create the threads using Runnable interface to perform the above operations.	**3**
3A.	Create a bill generating application for "**NammaBzzar**" customers. NammaBzzar categorizes the customers as: GoldCustomer & SilverCustomers. Each customer category has attributes – Customer id, Phone No, Customer Name, Price, Qty and a method to generate Bill which is to be implemented by the subclasses corresponding to each kind of customers. Bill amount for GoldCustomer is generated based on the special discount as 10% if the total amount of order placed is more than 5000/-. SilverCustomers will get 10 score points if the total amount of order placed is more than 1000 and allow straight discount of 50/-. Generate the bill for '**n**' customers based on the category of the customer entered (menu driven) by the user, display the Customer details, bill amount, discount/score points as per the above criteria. Implement the above using dynamic method dispatch.	**4**
3B.	Write a java program to calculate total monthly expenses of **n** persons by implementing *HouseHoldExpenses* and *OfficialExpenses* interfaces. Household expenses include: electricity bill, water bill, newspaper bill, grocery expenses. Official expenses include: bank loan EMI, transportation expense and mobile bill. Write a complete program to demonstrate these interfaces.	**4**
4A.	Develop a Java application to find the minimum distance and maximum distance between two neighboring a's in the given input sequence of characters. Display the positions of a's corresponding to the minimum and maximum distances.	**5**

Example



In the above example, the positions of two a's with minimum distance and maximum distance are: 17 & 19 and 10 & 17 respectively. Min_dist value is 2, Max_dist value is 7.

****