Reg. No.



MANIPAL INSTITUTE OF TECHNOLOGY

MANIPAL (A constituent unit of MAHE, Manipal)

## III SEMESTER B.TECH. (COMPUTER SCIENCE AND ENGINEERING) DEGREE GRADE IMPROVEMENT/MAKEUP EXAMINATIONS, JULY 2021 SUBJECT: OBJECT ORIENTED PROGRAMMING [CSE 2154] REVISED CREDIT SYSTEM (30/07/2021)

Time: 2 Hours

MAX. MARKS: 40

## **Instructions to Candidates:**

- Answer ANY FOUR FULL questions.
- Missing data may be suitably assumed.

1A. Consider a class Student with name, roll number and an array marks[] to store the marks of FIVE subjects as its private data members.

Include the following methods and constructors:

i) a display function to display the data members.

ii) a default constructor

iii) a constructor to initialize name and marks of the Student(Note: Each student has a unique roll number generated by the program).

iv) a member function Avg() to calculate average of five subject marks of a student and return the result. In main() create an array of students and initialize them appropriately and demonstrate use of all the above functions. (6)

1B. For finding a value in an array, following code is available. Does the program execute and generate correct output? If not why? Modify the code. (4) **public class** FindValue {

```
}
}
if (found)
{
    System.out.println("Found " + searchVal + " at index "+ i);
}
else
{
    System.out.println(searchVal + " not in the array ");
}
}
```

2A. Design a class Grading with the methods, get\_RegNo() and put\_RegNo() to read and display the Reg\_No of each student and get\_ Acad() and put\_Acad() to read and display their marks in 5 subjects(Maximum mark in each subject is 100). Declare the Reg\_No & marks as private data members. Create an interface called Co\_Curricular with get\_Score() to read the points scored in Co-curricular activities (Maximum score is 100) and methods put\_Score() that will display the score obtained by a student in Co\_Curricular activities. Design a class called Result that will implement the methods get\_Score() to put\_Score() and generate the final score (Maximum score is 100) based on the marks obtained from the Co-Curricular activities (20%) and the Academic score(80%) obtained from the superclass Student. Write a demo class to create an object of Result class, input the details and display the final score. (8)

```
2B. Suppose an interface MyConst is defined as follows:
package mypack;
public interface MyConst {
  public static final int COUNT = 42;
}
Demonstrate the two different ways to bring the constant COUNT into view for use in another class:
class MyClass {
  public static void main(String[] args) {
    System.out.println(COUNT);
  }
}(2)
```

3A. It is fine for the Java library to divide classes and interfaces into packages, but why do you need to do so for your programs? What is wrong with just giving all your classes unique names so that there is never a name conflict?(2)

3B. Create a method TransMat() to find the transpose of a square matrix. Place this in a package called pTrans. Let this package be present in a folder called "myPackages", which is a folder in your present working directory (eg:- D\OOP\mypackages\ pTrans). Write a main method in package pCheck to read a matrix. Using the methods TransMat() from package pTrans check if the entered matrix is symmetric or not. [Hint : a matrix A is symmetric if  $A=A^T$ ] (8)

4A. Write a Java program to create a Counter class with a private count instance variable and two methods. The first method increment() tries to increments count by 1. If count is already at its maximum of 3, then it waits until count is less than 3 before incrementing it. The other method decrement() attempts to decrement count by 1. If count is already at its minimum of 0, then it waits until count is greater than 0 before decrementing it. Every time either method has to wait, it displays a statement saying why it is waiting. Also, every time an increment or decrement occurs, the counter displays a statement that says what occurred and shows count's new value.

Demonstrates how two synchronized methods need to communicate to keep a counter between 0 and 3 when one thread repeatedly tries to increase the counter and the other thread repeatedly tries to decrease the counter. One thread class calls the Counter's increment() method 20 times. In between each call, it sleeps for a random amount of time between 0 and 500 milliseconds and the other thread calls the Counter's decrement() method 20 times. In between each call, it sleeps for a random amount of time between each call, it sleeps for a random amount of time between each call, it sleeps for a random amount of time between each call, it sleeps for a random amount of time between each call, it sleeps for a random amount of time between each call, it sleeps for a random amount of time between 0 and 500 milliseconds.

Write a CounterDemo class with a main() method that creates one Counter and the two threads and starts the threads running. Create just one thread class by implementing Runnable interface that either increments or decrements the counter, depending on a parameter passed to the thread class's constructor. Make sure that the main thread ends at the last. (6)

4B. Design a class ReserveTicket with an instance variable seatsavailable, and a method void reserve(int numberofseats). If `numberofseats' is greater than `seatsavailable' or `numberofseats' is less than 1 then throw an user defined exception SeatNotBookedException. in the main program one must be able to know why a seat was not booked. (4)

5A. Below is an attempt to make bubble sort a generic method so that it can sort all kinds of arrays,. Explain what happens when the following method is compiled? (2)

## public class CounterDemo

```
{ <T> void bubbleSort(T[] data)
 { for(int a=1; a < data.length; a++)
    for(int b=data.length-1; b >= a; b--)
      { if(data[b-1] > data[b])
      { // if out of order // exchange elements
            T x = data[b-1];
            data[b-1] = data[b];
            data[b] = x;
      }
    }
}
```

5B. Given the following code snippets, identify whether each one is compliant or noncompliant. If noncompliant, write the compliant solution and the CERT Oracle Secure Coding Standard for Java rule associated. (8)

Connection conn = null;

Statement stmt = null;

try{

conn = DriverManager.getConnection(DB\_URL,USER,PASS);

```
stmt = conn.createStatement();
```

// ...

}catch(Exception e){

e.printStackTrace();

}finally{

stmt.close();

conn.close();

```
}
```

//code snippet 2

long num1, num2, result;

result = num1 / num2;

//code snippet 3

```
BigInteger x = new BigInteger("530500452766");
```

byte[] byteArray = x.toByteArray();

String s = new String(byteArray);

6A. Create a class Account (acno, name, balance, pin) with a constructor and two methods: i) deposit to deposit the amount and ii) withDraw to withdraw the amount (this should check for a minimum balance of rs. 500). (2)

6B.Create a javafx class Atm to mimic the atm operations. The GUI is as shown in Figure 6.1 below. Use the grid layout and the pin entered should displayed as '\*\*\*\*' (i.e., it should be hidden). Create 3 objects of the Account class (Refer 6A) and use them in the atm operations. It should validate the entered acno, pin and balance and allow the operation for valid input only and the balance should be printed with the account holder's name. Otherwise it should print the corresponding message like wrong acno, wrong pin

and insufficient fund. Write a complete javafx program with import statements, controls as shown in fig and event handling. Once the application is running, it should allow to switch between the users and the operations. (8)

Account Operations		_		×
Account operations				
Enter Account No:	111			
Enter your Pin:	••••			
Select Operation:	Deposit		○ w	ithDraw
Enter Amount:	2000			
	Submit			
Anand Your Account Balance: 7000				

Figure 6.1 GUI