30 May, 2021

EAST WEST UNIVERSITY
Department of Computer Science and Engineering
B.Sc. in Computer Science and Engineering Program
Final Examination, Spring 2021 Semester

Course:	**CSE 110 Object Oriented Programming, Section-1**
Instructor:	**Tanni Mittra, Senior Lecturer, CSE Department**
Full Marks:	**35 (35 will be counted for final grading)**
Time:	**1 Hour and 30 Minutes**
Submission Time	**10 Minutes**

Note: There are **6** (**Six**) questions, answer ALL of them. Course Outcome (CO), Cognitive Level and Mark of each question are mentioned at the right margin.

1. Create an Abstract class ***DataReader*** with two Abstract methods ***ReadData(String filename) & WriteData()***. [CO3, C3, Mark: 8]

Create a class ***TextFile*** that will extend ***DataReader*** class with an instance array variable ***Filenumb[]*** of integer type . Override ***ReadData(String filename)*** method in such way that will read data from the text file mentioned in parameter of the function and store the data in the array. For this class consider that, ***ReadData(String filename)*** will read text file that contains only integer numbers written in separate lines. Again override ***WriteData()*** method to write the integer values of the array in a new textfile named ***"ExamTextFile.txt".***

Create another class ***BinaryFile*** that will extend ***DataReader*** class with an instance variable ***binarynum*** as ArrayList of integer type. This class override ***ReadData(String filename)*** method in such way that will read data from the binary file mentioned in parameter of the function and store the data in the ArrayList. Again override ***WriteData()*** to display the integer data of the ArrayList.

2. Define a Generic class named ***Calculator*** that has two instance variables: ***Input1*** and ***Input2***. Where each of the input can take objects of primitive data type classes. The partial class diagram of ***Calculator*** class is given below. [CO3, C3, Mark: 5]

Calculator <T>
- Input1: T - Input2: T
+Calculator(T num1, T num2) +DisplayResult(): void + Add(): T + Subtract():T +Multiply():T +Division():T

Complete this class definition by including all the methods. DisplayResult() method will display the result of addition, subtraction, multiplication and division of two numbers. Write a main method that create two objects of Calculator class of Integer & Double type. Performs the Calculator operations and displays the output.

3. Objects of the ***Calculator*** class as defined in question 2, must have non-negative ***Input1*** and ***Input2***. If negative values are given for any fields, a user-defined exception NegativeValueException must be thrown and the program must stop executing. [CO3, C3, Mark: 5]

Define a user-defined exception class NegativeValueException.

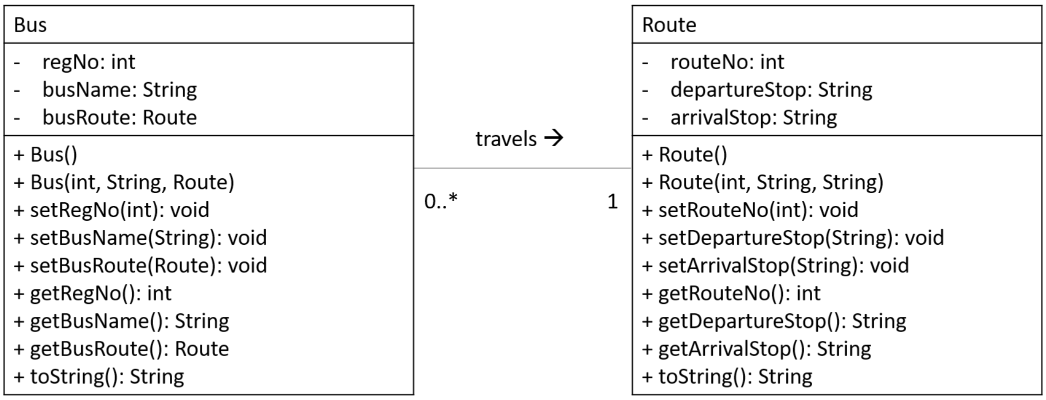
Then, modify the ***Calculator*** class such that if the ***Input1*** and ***Input2*** is negative, the class must throw a NegativeValueException. Write only the part of the class that you have modified. No need to write the full definition again.

Write only the part of a main method within the Main class that creates a Calculator type object, take input of two instance variables surrounded with appropriate try-catch block.

4. Create a ***Vehicle*** class that has instance variables vehiclename, mileage, and capacity, along with appropriate constructors, accessors, and mutators. Make the ***Vehicle*** class implement the Comparable interface. Define the compareTo method to order ***Vehicle*** objects based on the mileage. In the main method, create an array of at least five ***Vehicle*** objects, sort those using Arrays. Sort, and output the ***Vehicle***. They should be listed by ascending mileage. [CO3, C3, Mark: 6]

Next, modify the compareTo method so it orders ***Vehicle*** objects based on the lexicographic ordering of their vehicle name. The program should now output the vehicles ordered by name.

5. Consider the following class diagram. [CO3, C3, Mark: 6]



.

Figure 1: A class diagram

Assume that the above-mentioned classes have already been implemented. Also consider that you have a Main class and within the main () method in which the following array lists of Route and Bus type objects are created.

```
ArrayList<Route> routes = new ArrayList<>();
ArrayList<Bus> buses = new ArrayList<>();
// adding few routes and buses
routes.add(……);
buses.add(……);
```

a) **Complete** the above program in Java that writes all Bus type objects as created and held in *buses* ArrayList into a file named **“buses.dat”** using **DataOutputStream** class. You must write the data in the following format.

```
    regNo busName routeNo departureStop arrivalStop
```

b) Do you need to change the class definition of the above-mentioned classes to write objects of those classes directly into a file using **ObjectOutputStream** class? If yes, what would be the change?

6. Write a java program that will create two threads. Where one thread will perform add operation of two numbers and other thread will perform subtraction operation of two numbers. But you have to ensure that addition thread will run first and subtraction thread will run later. Write a main method to run the two threads. [CO3, C3, Mark: 5]