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

Course:	**CSE 110 (Object Oriented Programming), Section – 6**
Instructor:	**Yeasir Rayhan, Lecturer, CSE**
Full Marks:	**32 (32 will be counted for final grading)**
Time:	**2 Hour + 10 minutes for submission**

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

---

1. Consider the following class named `Train` with a predefined capacity and an array to keep track of the status of the seats (whether a seat is empty or occupied). When seat[i] = 0, the i-indexed seat is empty and when seat[i] = 1, it is occupied. One can book a seat in either of the two ways: [CO3, C3, Mark: 5]

i) One can specify the seat no he wants to book: `int bookSeat(int seatNo)`. If the seat is already taken an exception will be triggered specifying the seat number he is trying to book is already taken, otherwise the seat will be his.

ii) One can simply book any seat that is available: `int bookSeat()`. If the train is not full to its capacity, he will be given the first seat that is available and if not, an exception will be triggered specifying that the train is already full.

Implement appropriate custom exceptions and the overloaded `bookSeat` methods of class Train below.

```java
public class Train {
    int capacity;
    int seat[];

    public BookTicket(int capacity) {
        this.capacity = capacity;
        this.seat = new int[capacity];
    }
    public int bookSeat(int seatNo){
        //implement this method
    }

    public int bookSeat(){
        //implement this method
    }
}
```

2. Consider the `main` method of class `MultipleThreads.` For an account named `account1` a transaction is requested: deposit an amount of 500 to `account1`. [CO3, C3, Mark: 4]

Implement class `TransactionDeposit` so that the following program outputs:
Balance of account1 = 1500

```java
class MultipleThreads{
    public static void main(String args[]){
        Account account1 = new Account();
        account1.balance = 1000;
        TransactionDeposit t1 = new TransactionDeposit(account1,
                                                    500);
        t1.t.join();
        System.out.println("Balance of account1 = " +
                                                    account1.balance);
    }
}
```

3. **Write** a generic method to count the number of elements in a `Vector` collection that are even. [CO3, C3, Mark: 3]

4. **Create** 5 dummy Item objects of class Item and **Implement** the buttonAction method of the following ListViewController class that loads the names of these 5 dummy objects to a list view. And when any of the element of the list view is selected, an alert is shown with the price of that item. [CO3, C3, Mark: 5]

```
class Item{
    int id;
    String name;
    double price;

    public Item(int id, String name, double price) {
        this.id = id;
        this.name = name;
        this.price = price;
    }
}

public class ListViewController {
    @FXML
    private ListView<String> listView;
    @FXML
    private Button button;

    @FXML
    void buttonAction(ActionEvent event) {

    }
}
```

5. (a) **Write** a line of code that instantiates an `ArrayList` object named `labels` that can hold elements (and only `Labels`) of class `Label` with an initial capacity of 40 and does not produce any compiler errors or warnings. [CO3, C3, Mark: 3]

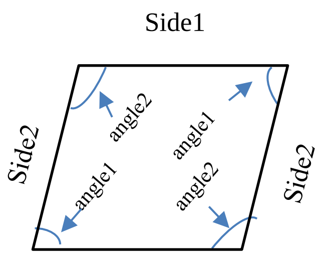
(b) Continuing from the previous question, **write** a `for-each` loop that prints to the console the the text of each Label in the labels that is not disabled. Assume `Label` has `String getText()` and `boolean isDisabled()` methods.

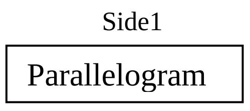
(c) **Repeat** the same task in (b) using an `arraylist iterator`.

```
class Label{
    boolean disabled;
    String text;
    public String getText(){
        return text;
    }
    public boolean isDisabled(){
        return disabled;
    }
}
```

6. A parallelogram is a *polygon with 4 sides (vertices)* whose opposite sides are parallel and equal, therefore opposite angles are also equal. [CO2, C3, Mark: 6]
A *rhombus* is a parallelogram with all the sides and all the angles equal.
A *rectangle* is a parallelogram with opposite sides equal and all angles are right (90) angled
A *square* is a rectangle with all sides equal.

Implement the empty constructors and abstract method of the following classes: `Parallelogram`, `Rhombus`, `Rectangle` and `Square` given below. Note that you cannot add any extra constructor in any of the classes.





```
abstract class Parallelogram{
double side1;
    double side2;
    double angle1;
    double angle2;
    public Parallelogram(double side1, double side2, double angle1,
           double angle2) {

    }
    public abstract getArea();

}
class Rhombus extends Parallelogram{
    public Rhombus(double side1, double angle1) {

    }
}
class Rectangle extends Parallelogram{
    public Rectangle(double side1, double side2) {

    }
}
class Square extends Rectangle{
    public Square(double side1) {

    }
}
```

7. In this question, you will model some celestial bodies, including planets and stars. [CO2, C3, Mark: 6]

a. The `Planet` constructor accepts a radius, which should set the corresponding data member. In addition to the constructor, the Planet class should have an abstract method `calcAtmosphereThickness()` that returns the thickness of the Planet's atmosphere (double). **Implement** these two methods.

```
abstract class Planet {
    double radius;
}
```

b. There are two kinds of planet: `RockyWorld` and `GasGiant`. For modeling purposes, we assume that the atmospheric thickness on a `RockyWorld` is 1% of its planetary radius (while on a `GasGiant` it is 80% of its planetary radius). For modeling purposes, we assume that the atmospheric thickness on a `RockyWorld` is 1% of its planetary radius. Complete the `RockyWorld` class below while inheriting as much as possible from its parent class and implementing any necessary methods.

```
class RockyWorld extends Planet {

}
```