National University

Of Computer & Emerging Sciences Peshawar Campus

Lab 07: Task

Design the Class Diagram of all question and Implement in Java.

QN: 01 Polymorphism

Vehicles can be classified according to what energy source powers them. The program you are given has 3 vehicle classes: Vehicle, ElectricVehicle and HybridVehicle.
ElectricVehicle and HybridVehicle classes are inherited from Vehicle class. Complete the program by reimplementing method resource() in inherited classes, so that the given calls work correctly.

The subclass can implement a parent class method based on its requirement.

```java
class Main {
   public static void main(String[] args) {

       Vehicle vehicle = new Vehicle();
       Vehicle electric = new ElectricVehicle();
       Vehicle hybrid = new HybridVehicle();
       //calls
       vehicle.start();
       vehicle.resource();
       electric.start();
       electric.resource();
       hybrid.start();
       hybrid.resource();
   }
}
class Vehicle{
    public void start(){
        System.out.println("Starting");
    }
    public void resource(){
        System.out.println("I use petrol");
    }
}
class ElectricVehicle extends Vehicle{
    /*reimplement resource() method
    to output "I use electricity"*/
}

class HybridVehicle extends Vehicle{
    /*reimplement resource() method
    to output "I use both petrol and electricity"*/
}
```

QN: 02 Overriding

You are writing a program that calculates interest income based on account type. If it is a saving account, 20% is added to the balance in the account. If it is a checking account, 5% more is added to the balance in the account.
The program you are given should the balance of the savings and checking accounts and output the appropriate balance plus interest income
Account class and SavingAcc & CheckingAcc classes inherited from it are already defined.
Override getIncome() methods in the inherited classes to calculate and return the account balances, so that the given outputs work correctly.

Sample Input
100.0
100.0

Sample Output
120.0
105.0

Explanation
The first input represents balance in the savings account, so the balance plus interest income should be 100.0 + 100.0*0.2 = 120.0.
The second input represents balance in the checking account, so the balance plus interest income should be 100.0 + 100.0*0.05 = 105.0.
You should use the percentage in your formula as a double number.

Given Code:

Lab 07: Task

```
import java.util.Scanner;
class Main {
   public static void main(String[] args) {
       Scanner read = new Scanner(System.in);
       double saving = read.nextDouble();
       double checking = read.nextDouble();

       Account savingAcc = new SavingAcc(saving);
       Account checkingAcc = new CheckingAcc(checking);

       System.out.println(savingAcc.getIncome());
       System.out.println(checkingAcc.getIncome());
   }
}
class Account {

   private double amount;

   public Account(double amount) {
       this.amount = amount;
   }

   public double getAmount() {
       return amount;
   }

   public double getIncome() {
       return 0;
   }
}

class SavingAcc extends Account {

   public SavingAcc(double amount) {
       super(amount);
   }
   //Override the method for saving account
   public double getIncome() {

   }
}
class CheckingAcc extends Account {

   public CheckingAcc(double amount) {
       super(amount);
   }
```

Lab 07: Task

```
    //Override the method for checking account
    public double getIncome() {

    }
}
```

QN: 03 Shapes

You are working on a graphical app, which includes multiple different shapes. The given code declares a base **Shape** class with an abstract **area**() method and a **width** attribute.
You need to create two **Shape** subclasses, **Square** and **Circle**, which initialize the width attribute using their constructor, and define their **area**() methods.
The **area**() for the **Square** class should output the area of the square (the square of the width), while for the **Circle**, it should output the area of the given circle (PI*width*width).
The code in main creates two objects with the given user input and calls the area() methods.

Sample Input:
5
2

Sample Output:
25
12.566370614359172

The area of the square is 5*5=**25**, while the area of the circle is PI*2*2=**12.566370614359172**

Use the **Math.PI** constant for the area calculation of the circle.

```
import java.util.Scanner;

abstract class Shape {
    int width;
```

Lab 07: Task

```
    abstract void area();
}
//your code goes here


public class Program {
    public static void main(String[ ] args) {
        Scanner sc = new Scanner(System.in);
        int x = sc.nextInt();
        int y = sc.nextInt();

        Square a = new Square(x);
        Circle b = new Circle(y);
        a.area();
        b.area();
    }
}
```

QN: 04 Realtime Example of Relationship in Java

A most common example of ______ relationship is "A student has an address". A student has many pieces of information such as name, roll no, email id, etc.

It also contains one more important object named "address" that contains information such as city, state, country, zip code.

Let's implement this common example programmatically to understand the __________ relationship in java.

In this example program, Student class has an object of Address class where the address object contains its own information such as city, state, country, etc.

This relationship is Student Has-A address and is called ______?.

Lab 07: Task

Define what type of relationship exist here?

QN: 05 Realtime Example of Relationship in Java

If the car is destroyed then its engine will be destroyed as well. So, without the existence of car, there is no life of an engine. The life of an engine is totally dependent on the life cycle of car.

Let’s write it programmatically with the help of an example program.

- **Engine attributes are horsepower and type**
- **Car attributes are Name and Engine.**
- **Define what type of relationship exist here.**