Part 3

Github link: https://github.com/kalp104/java-part-3.git

Question 1	Create an abstract class GeometricObject as the superclass for Circle and Rectangle. GeometricObject models common features of geometric objects. Both Circle and Rectangle contain the getArea() and getPerimeter() methods for computing the area and perimeter of a circle and a rectangle. Since you can compute areas and perimeters for all geometric objects, so define the getArea() and getPerimeter() methods in the GeometricObject class. Give implementation in the specific type of geometric object. Create TestGeometricObject class to display area and perimeter of Rectangle and Triangle, compare area of both and display results. Design of all classes are given in the following UML diagram.

```
abstract public class GeometricObject {
    public int Radius;
    public String color;
    public boolean filled;
    public int length;
    public int width;

    public GeometricObject(String color, boolean filled) {
        this.color = color;
        this.filled = filled;
    }


    public void getcolor(String color,boolean filled){
        this.color = color;
        this.filled = filled;
    }
    public String string()
    {
       return color;
    }
    public boolean filled()
    {
        return filled;
    }

    public void getradius(int r){
        Radius = r;
    }
     public double Area(){
         return 0;
     }
     public double perimeter() {
```

```
         return 0;
     }
}
class circle extends GeometricObject
        {
            public circle(String color, boolean filled,int
radius) {
                super(color, filled);
                Radius = radius;
            }

//            public void getradius(int r){
//                Radius = r;
//            }
            public double Area() {
                return (3.14 * (Radius * Radius));
            }
            public double perimeter(){
                return (3.14 * (2 * Radius));
            }


            public static void main(String[] args)
            {
            circle obj = new circle("red",true,4);
            ractangle obj1 = new
ractangle("blue",true,7,8);
//            obj.getcolor("red",true);
//            obj.getradius(4);
            double area = obj.Area();
            double perimeter = obj.perimeter();

                double area1 = obj1.Area();
                double perimeter1 = obj1.perimeter();
            System.out.println( " availability : "+
obj.filled() + "the color is : " + obj.string());
            System.out.println("Area of circle is : " +
area );
            System.out.println("perimeter of circle : " +
perimeter);

                System.out.println( " availability : "+
obj.filled() + "the color is : " + obj1.string());
                System.out.println("Area of ractangle is :
" + area1 );
                System.out.println("perimeter of ractangle
: " + perimeter1);
            }

        }
    public ractangle(String color,boolean filled,int
length,int width)
    {
        super(color,filled);
        this.color = color;
        this.filled = filled;
        this.length = length;
        this.width = width;
    }
    public double Area(){
        return (length*width);
```

```
    }
    public double perimeter() {
        return (2*(length+width));
    }


}
```

Question 2	Write a program to create a default method in an interface IPrinter. Create an interface IPrinter and IScanner. You can assume variables and methods for both interfaces. Create a concrete class to implement both the interfaces. Create 5 objects of the class, store it in Vector and display the result of the vector.

```
public interface IPrinter {
    public int vector1=10;
    void display();
}
```

```
public interface IScanner extends IPrinter{
    public int vector2 = 19;
    void display1();
}
```

```
public class question2 implements IScanner{

    @Override
    public void display() {
        System.out.println("in IPrinter");
    }
    @Override
    public void display1()
    {
        System.out.println("in IScanner");
    }

    public static void main(String[] args) {
        question2 obj = new question2();
        obj.display();
        obj.display1();
        System.out.println("the value of vector1 is " +
vector1);
        System.out.println("the value of vector2 is " +
vector2);
    }
}
```

Question-3	WAP that illustrate the interface inheritance. Interface P is extended by P1 and P2 interfaces. 1,2 Interface P12 extends both P1 and P2. Each interface declares one method and one constant. Create one class that implements P12. By using the object of the class invokes each of its method and displays constant.

```
public interface interface_P {
    final int vector = 10;
    void display();
}
```

```java
public interface interface_P1 extends interface_P{
    final int vector1 = 11;
    void display1();
}
```

```java
public interface interface_P2 extends interface_P{
    final int vector2 =12;
    void display2();
}
```

```java
public interface interface_P12 extends
interface_P1,interface_P2{
    final int vector3=13;
    void display3();
}
```

```java
public class main_class implements interface_P12{
    @Override
    public void display() {
        System.out.println("we are in interface_p");
    }

    @Override
    public void display1() {
        System.out.println("we are in interface_p1");
    }

    @Override
    public void display3() {
        System.out.println("we are in interface_P2");
    }

    @Override
    public void display2() {
        System.out.println("we are in interface_p12");
    }

    public static void main(String[] args) {
        main_class obj = new main_class();
        obj.display();
        obj.display1();
        obj.display2();
        obj.display3();
        System.out.println("interface_p vector = " +
obj.vector);
        System.out.println("interface_p1 vector = " +
obj.vector1);
        System.out.println("interface_p2 vector = " +
obj.vector2);
        System.out.println("interface_p12 vector = " +
obj.vector3);
    }
}
```

Question 4:	Develop a Program that illustrate method overriding concept.

```
public class bank {
    public int rate_of_interest() {
        return 0;
    }
}
```

```
public class SBI extends bank {
       public int rate_of_interest(){
        return 8;
        }
}
```

```
public class ICICI extends bank {
    @Override
    public int rate_of_interest() {
        return 7;
    }
}
```

```
public class BOB extends bank {
    @Override
    public int rate_of_interest() {
        return 9;
    }
}
```

```
public class method_overriding {
    public static void main(String[] args) {
        // method overriding
        SBI obj = new SBI();
        ICICI obj1 = new ICICI();
        BOB obj2 = new BOB();

        System.out.println("interest rate information : ");
        System.out.println("SBI bank's interest rate is : "
+ obj.rate_of_interest());
        System.out.println("ICICI bank's interest rate is :
" + obj1.rate_of_interest());
        System.out.println("BOB bank's interest rate is : "
+ obj2.rate_of_interest());
    }
}
```

Question 5	Write a java program which shows importing of classes from other user define packages.

```
package usingpackage;
public class useofpackage {
    public void demo(){
        System.out.println("this is practical 6th ");
    }
}
```

```java
import usingpackage.*;
public class demmoclass {
    public static void main(String[] args) {
        useofpackage obj = new useofpackage();
        obj.demo();
    }
}
```

Question 6

Write a program that demonstrates use of packages & import statements.

```java
package myclassadition;
public class adition {
    public int aditionof(int a,int b){
        return a+b;
    }
}
```

```java
package myclassdivision;
public class division {
    public float divisionof(float a,float b)
    {
        return a/b;
    }
}
```

```java
package myclassmodulo;
public class modulo {
    public int moduloof(int a,int b)
    {
        return a%b;
    }
}
```

```java
package myclassmultiplication;
public class multiplication {
    public int multiplicationof(int a, int b) {
        return a*b;
    }


}
```

```java
package myclasssubtraction;
public class subtraction {
    public int subtractionof(int a,int b){
        return a-b;
    }
}
```

```java
import myclassadition.*;
import myclassdivision.*;
import myclassmultiplication.*;
import myclasssubtraction.*;
import myclassmodulo.*;

public class mainclass {
    public static void main(String[] args) {
        adition obj = new adition();
```

```
        System.out.println("addition is : " +
obj.aditionof(10,20));
        subtraction obj1 = new subtraction();
        System.out.println("subtraction is : " +
obj1.subtractionof(20,10));
        multiplication obj2 = new multiplication();
        System.out.println("multiplication is : " +
obj2.multiplicationof(10,20));
        division obj3 = new division();
        System.out.println("division is : " +
obj3.divisionof(20,10));
        modulo obj4 = new modulo();
        System.out.println("modulo is : " +
obj4.moduloof(50,6));
    }
}
```

Question 7	Write a program that illustrates the significance of interface default method.

```
interface TestInterface{
    public void square(int a);

    // default method
    default void show()
    {
        System.out.println("Default Method Executed");
    }
}
```

```
// abstract method


class TestClass implements TestInterface
{
    public void square(int a)
    {
        System.out.println(a*a);
    }

    public static void main(String args[])
    {
        TestClass d = new TestClass();
        System.out.print("the square of given number is :
");
        d.square(4);
        d.show();

// default method executed d.show();
    }
}
```