Name : Hardik Patel
Std Id : 202103032

1. WAP in JAVA where, a Shape Abstract Class and 3 subclasses, Triangle,Rectangle, and Circle. Shape defines two int variables: x and y, which represent the coordinates of the center of the shape. Shape also defines 2 abstract methods calculateArea() and calculateCircumference().

Code :

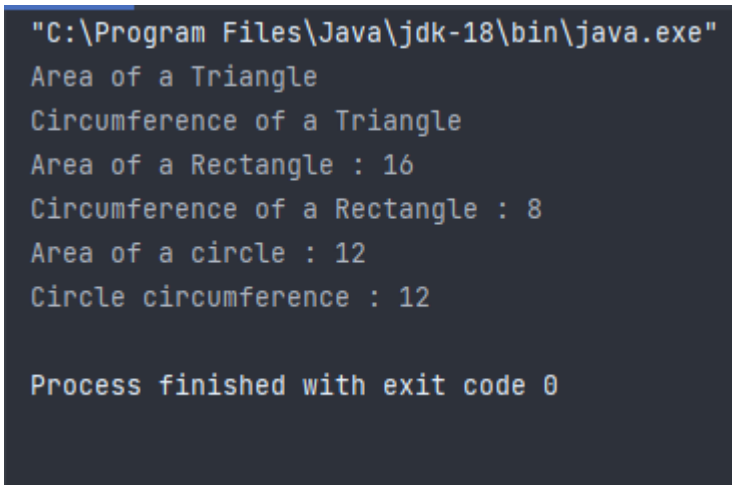
```
package com.OOP.Assignment8;
abstract class Shape {
    // coordinate for center of the shapes
    int x;
    int y;
    //abstract method
    abstract void calculateArea();
    abstract void calculateCircumference();
    public Shape(int x,int y) {
        this.x = x;
        this.y = y;
    }
}
class Triangle extends Shape{
    public Triangle(int x, int y) {
        super(x, y);
    }
    @Override
    void calculateArea() {
        System.out.println("Area of a Triangle " );
    }
    @Override
    void calculateCircumference() {
        System.out.println("Circumference of a Triangle ");
    }
}
class Rectangle extends Shape{
    public Rectangle(int x, int y) {
        super(x, y);
    }
    @Override
    void calculateArea() {
        System.out.println("Area of a Rectangle : " + 4*x*y);
    }
    @Override
```

```
    void calculateCircumference() {
        System.out.println("Circumference of a Rectangle : " +
2*x*y);
    }
}

class Circle extends Shape{
    public Circle(int x, int y) {
        super(x, y);
    }
    @Override
    void calculateArea() {
        System.out.println("Area of a circle : " + (22*x*x)/7);
    }
    @Override
    void calculateCircumference() {
        System.out.println("Circle circumference : " +
(2*22*x)/7);
    }
}

public class Main {
    public static void main(String[] args) {
        Shape t = new Triangle(2,2);
        t.calculateArea();
        t.calculateCircumference();
        Shape r = new Rectangle(2,2);
        r.calculateArea();
        r.calculateCircumference();
        Shape c = new Circle(2,2);
        c.calculateArea();
        c.calculateCircumference();
    }
}
```

Output :



2. State whether the following code snippet is right/wrong. Justify your answer.

A)
.

```
abstract class Language {
public void display() {
System.out.println("This is Java Programming");
}
}
class Main extends Language {

public static void main(String[] args) {
Main obj = new Main();
obj.display();
}

}
```

Ans : Above code is Right
Though abstract classes cannot be instantiated, we can create subclasses from it. We can then access members of the abstract class using the object of the subclass.

Jstify : In above code, we have created an abstract class named Language. The class contains a regular method display().

We have created the Main class that inherits the abstract class. Here obj is the object of the child class Main. We are calling the method of the abstract class using the object obj.

B)

```
abstract class Bike{
void run();
}
class Honda extends Bike{
void run(){System.out.println("running safely");}

public static void main(String args[]){
Bike obj = new Honda4();
obj.run();
}

}
```

Ans : Above code snippet is Wrong

Justify : In above code we are making new object with wrong class name
it must be new Honda() instead of Honda4()

3.Banks are supposed to implement their own method that returns back the rate of interest for personal loans. To implement this scenario in SBI and PNB banks they are supposed to extend the Bank class provided by the RBI that has a single abstract method called getRateOfInterest() that returns back an integer. WAP in JAVA where in the banks BankA and BankB that have their own classes as BankA and BankB that extend from the BANK class and implement the abstract method.

Code :

```
package com.OOP.Assignment8;
public abstract class Bank{

    public abstract int getRateOfInterest();

    public static void main(String[] args) {

        Bank bank1=new SBI();
        Bank bank2=new PNB();

        System.out.println("Rate of Interest in SBI is:
"+bank1.getRateOfInterest()+" %");
        System.out.println("Rate of Interest in PNB is:
"+bank2.getRateOfInterest()+" %");
    }
}
```

```
class SBI extends Bank{
    public int getRateOfInterest(){
        return 8;
    }
}
class PNB extends Bank{
    public int getRateOfInterest(){
        return 6;
    }
}
```

Output :

```
"C:\Program Files\Java\jdk-18\bin\java.exe
Rate of Interest in SBI is: 8 %
Rate of Interest in PNB is: 6 %

Process finished with exit code 0
```

4. What is the difference between Abstract Class and Interface with an example.

Ans : We can't extends multiple abstract classes but we can implement multiple interfaces at a time

The access specifiers used with classes are private, protected, and public , In Interface only Public.

We can not modify properties in interfaces as they are final

```
package com.OOP.Assignment8;
//statement 2
interface Bicycle {
    //statement 3
    // in interface properties are by default final
    final int s = 100;
    void applyBrake(int decrement);
    void speedUP(int increment);
}

interface Car{
    void gear(int gear);
}
```

```java
//statement 1
class Hero implements Bicycle,Car{
    int speed = 8;

    public void applyBrake(int decrement) {
        speed = speed - decrement;
        System.out.println("decremnted speed = " + speed);
    }

    public void speedUP(int increment) {
        speed = speed + increment;
        System.out.println("incremented speed = " + speed);
    }

    @Override
    public void gear(int gear) {
        System.out.println("gear = " + gear);
    }
}

public class Vehicle{

    public static void main(String[] args) {
        Hero hero = new Hero();
        hero.applyBrake(2);
        hero.speedUP(5);
        hero.gear(3);
        // we can not modify the properties in interface as
they are final
      //  hero.s = 101;
        System.out.println("main speed : " + hero.s);
    }
}
```

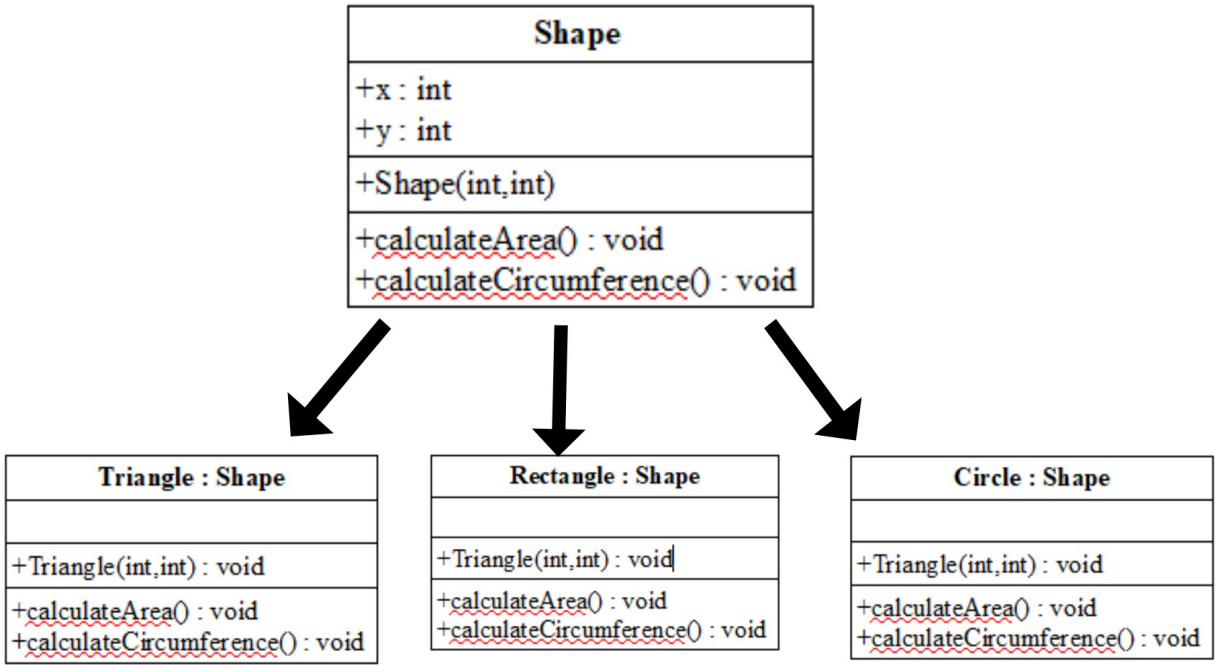
output :

```
"C:\Program Files\Java\jdk-18\bin\java.exe" "
decremnted speed = 6
incremented speed = 11
gear = 3
main speed : 100

Process finished with exit code 0
```

5. Create Class diagram for Q1 and Q3 of this assignment.

Q-1)



Q-3)class diagram for question 3