## Task No. 1: Write a program for exam department which provide abstract class and method of Exam type which contains general methods related to exams and can be used by different department for conducting exams.

**Solution:**

package Javalabtasks; public class main{

public static void main(String[] args) { Mathexam exam1=new Mathexam(); Scienceexam exam2=new Scienceexam(); exam2.conductexam(); exam2.passedstudents(21, 25); exam2.checkResult(); System.out.println("\n ");

exam1.conductexam(); exam1.passedstudents(20, 23); exam1.checkResult();

}

}

## Exam CLASS:

package Javalabtasks;

abstract class Exam {

abstract void conductexam();

public void checkResult(){ System.out.println("Checking results…");

}

abstract void passedstudents(double a,double b);

}

## Mathexam CLASS:

package Javalabtasks;

public class Mathexam extends Exam{ public void conductexam(){

System.out.println("Math Exam is Conducted");

}

public void passedstudents(double a,double b){ double c=a/b\*100;

System.out.println(c+" percent Students passed in Math Exam");

}

public void checkResult(){

super.checkResult();

}

}

## Science exam CLASS:

package Javalabtasks;

public class Scienceexam extends Exam{ public void conductexam(){

System.out.println("Science Exam is Conducted");

}

public void passedstudents(double a,double b){ double c=(a/b)\*100;

System.out.println(c+" percent students passed in Science Exam");

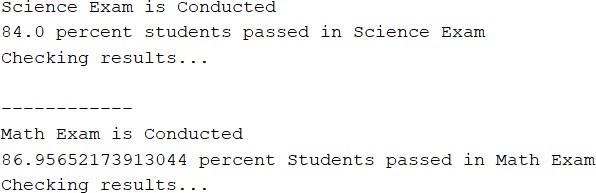
}

public void checkResult(){ super.checkResult();

}

}

# Output:

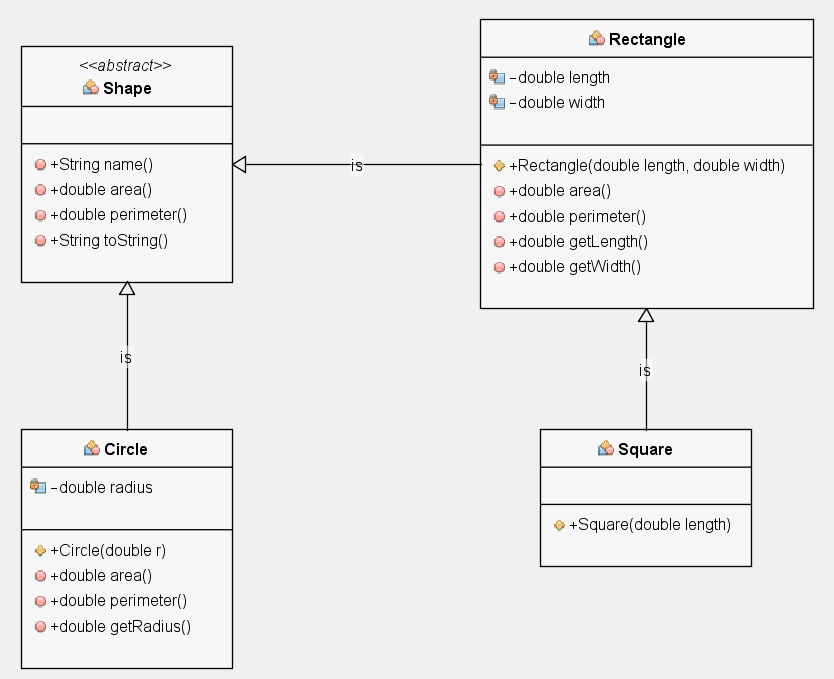


## Task No. 2 : You have to implement the UML diagram given below. Also Design and implement a subclass “EquilateralTriangle” having a double variable side denoting the three sides of the equilateral triangle [Note that since all the 3 sides are equal, the constructor will have only one parameter]. The area and perimeter of the equilateral triangle are given as follows:

**Area = ¼\* 3 \*(*side*)2 Perimeter = 3\**side***

## Provide accessor methods for the sides. Test your class using the TestShapes and

**DownCastingShapes classes.**



**Solution:**

**DownCastingShapes CLASS:**

package Javalabtasks;

public class DownCastingShapes{

public static void main(String[] args) {

Shape[] randomShapes = TestShapes.createShape();

for(int i = 0; i < randomShapes.length; i++){ System.out.println(randomShapes[i]); if(randomShapes[i] instanceof Circle)

System.out.println("Radius="+((Circle)randomShapes[i]).getRadius()); else if(randomShapes[i] instanceof Square) System.out.println("Length="+((Square)randomShapes[i]).getLength()); else if(randomShapes[i] instanceof Rectangle) System.out.println("Length="+((Rectangle)randomShapes[i]).getLength()+

"\nWidth= " +((Rectangle) randomShapes[i]).getWidth());

else if(randomShapes[i] instanceof EquilateralTriangle) System.out.println("Side="+((EquilateralTriangle)randomShapes[i]).getside());

}

}

}

## Test Shapes CLASS:

package Javalabtasks; import java.util.\*;

public class TestShapes {

public static Shape[] createShape() { final int SIZE = 5;

final double DIMENSION = 100; final int NUMBEROFSHAPES = 4;

Random generator = new Random();

//create an array having b/w 1 and SIZE entries

Shape[] randomShapes = new Shape[generator.nextInt(SIZE) + 1];

for(int i = 0; i < randomShapes.length; i++)

{

//randomly generate values b/w 0 and NUMBEROFSHAPES - 1 int assigner = generator.nextInt(NUMBEROFSHAPES);

switch(assigner) { case 0:

randomShapes[i] = new Rectangle(generator.nextDouble()\*DIMENSION,generator.nextDouble()\*DIMENSION);

break; case 1:

randomShapes[i] = new Circle(generator.nextDouble()\*DIMENSION);

break;

case 2:

randomShapes[i] = new Square(generator.nextDouble()\*DIMENSION);

break;

case 3:

randomShapes[i] = new EquilateralTriangle(generator.nextDouble()\*DIMENSION);

break;

}

}

return randomShapes;

}

public static void main(String[] args) {

Shape[] randomShapes = TestShapes.createShape(); for(int i = 0; i < randomShapes.length; i++)

System.out.println(randomShapes[i].toString());

}

}

## Shape CLASS:

package Javalabtasks;

abstract class Shape { public String name(){

return getClass().getSimpleName();

}

public abstract double area(); public abstract double perimeter();

public String toString() {

return "\n" +name() +"\n Area=" +area() +"\nPerimeter=" +perimeter();

}

}

## Rectangle CLASS:

package Javalabtasks;

public class Rectangle extends Shape{ private double length;

private double width;

public Rectangle(double length, double width){ this.length = length;

this.width = width;

}

public double area(){

return length \* width;

}

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

}

public double getLength(){ return length;

}

public double getWidth(){ return width;

}

}

## Square CLASS:

package Javalabtasks;

public class Square extends Rectangle{ public Square(double length){

super(length, length);

}

}

## Circle CLASS:

package Javalabtasks;

public class Circle extends Shape{ private double radius;

public Circle(double r){ radius = r;

}

public double area(){

return Math.PI \* (radius \* radius);

}

public double perimeter(){ return 2.0 \* Math.PI \* radius;

}

public double getRadius(){ return radius;

}

}

## Equilateral Triangle CLASS:

package Javalabtasks;

public class EquilateralTriangle extends Shape{ private double side;

public EquilateralTriangle(double side) { this.side = side;

}

public double area(){

return 0.25\*1.73\*side\*side;

}

public double perimeter(){ return 3\*side;

}

public double getside(){ return side;

}

}

# Output:

