# SD2 - Worksheet 3 - 8%

GRIFFITH COLLEGE DUBLIN

| | | | | | |
|---|---|---|---|---|---|
| **Student name:** | **Serianu Andrei-Silviu** | | | | |
| **Student number:** | **3144757** | | | | |
| **Faculty:** | **Computing Science** | | | | |
| **Course:** | **BSCH/BSCO/EXCH** | | **Stage/year:** | **2** | |
| **Subject:** | **Software Development 2** | | | | |
| **Study Mode:** | Full time | ☑ | Part-time | | |
| **Lecturer Name:** | **Gemma Deery** | | | | |
| **Assignment Title:** | **Worksheet 3** | | | | |
| Date due: | **19.03.2025** | | | | |
| Date submitted: | **19.03.2025** | | | | |

**Plagiarism disclaimer:**

*I understand that plagiarism is a serious offence and have read and understood the college policy on plagiarism. I also understand that I may receive a mark of zero if I have not identified and properly attributed sources which have been used, referred to, or have in any way influenced the preparation of this assignment, or if I have knowingly allowed others to plagiarise my work in this way.*

*I hereby certify that this assignment is my own work, based on my personal study and/or research, and that I have acknowledged all material and sources used in its preparation. I also certify that the assignment has not previously been submitted for assessment and that I have not copied in part or whole or otherwise plagiarised the work of anyone else, including other students.*

**Signed: _____Serianu_____ Date: ____19.03.2025_____**

Please note: **Students** MUST **retain a hard / soft copy of** ALL assignments as well as a receipt issued and signed by a member of Faculty as proof of submission.

**Repo Link:**

# Task 1

## Part 1

```java
package griffith;

abstract class Shape {
    private String name;

    public Shape(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

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

    @Override
    public String toString() {
        return "Shape: " + name;
    }
}
```

```java
package griffith;

class Circle extends Shape {
    private double radius;

    public Circle(String name, double radius) {
        super(name);
        this.radius = radius;
    }

    @Override
    public double area() {
        return Math.PI * radius * radius;
    }

    @Override
    public double perimeter() {
        return 2 * Math.PI * radius;
    }

    @Override
    public String toString() {
        return super.toString() + ", Radius: " + radius;
    }
}
```

```java
package griffith;

class Rhombus extends Shape {
    private double diagonal1, diagonal2, side;

    public Rhombus(String name, double diagonal1, double diagonal2, double side) {
        super(name);
        this.diagonal1 = diagonal1;
        this.diagonal2 = diagonal2;
        this.side = side;
    }

    @Override
    public double area() {
        return (diagonal1 * diagonal2) / 2;
    }

    @Override
    public double perimeter() {
        return 4 * side;
    }

    @Override
    public String toString() {
        return super.toString() + ", Diagonals: " + diagonal1 + " & " + diagonal2 + ", Side: " + side;
    }
}
```

```java
package griffith;

class RightAngledTriangle extends Shape {
    private double base, height, hypotenuse;

    public RightAngledTriangle(String name, double base, double height, double hypotenuse) {
        super(name);
        this.base = base;
        this.height = height;
        this.hypotenuse = hypotenuse;
    }

    @Override
    public double area() {
        return (base * height) / 2;
    }

    @Override
    public double perimeter() {
        return base + height + hypotenuse;
    }

    @Override
    public String toString() {
        return super.toString() + ", Base: " + base + ", Height: " + height + ", Hypotenuse: " + hypotenuse;
    }
}
```

## Part 2

```java
package griffith;

import java.util.*;

public class ShapeTest {
    public static void main(String[] args) {
        ArrayList<Shape> shapes = new ArrayList<>();
        shapes.add(new Circle("Circle", 3.5));
        shapes.add(new Circle("Circle", 5.0));
        shapes.add(new Rhombus("Rhombus", 4, 6, 5));
        shapes.add(new Rhombus("Rhombus", 6, 8, 7));
        shapes.add(new RightAngledTriangle("Triangle", 3, 4, 5));
        shapes.add(new RightAngledTriangle("Triangle", 6, 8, 10));

        for (Shape shape : shapes) {
            System.out.println(shape);
            System.out.println("Area: " + shape.area());
            System.out.println("Perimeter: " + shape.perimeter());
        }
    }
}
```

```java
package griffith;


import static org.junit.jupiter.api.Assertions.*;
import org.junit.jupiter.api.Test;

public class CircleTest {
    @Test
    void testCircleArea() {
        Circle circle = new Circle("Circle", 3.5);
        assertEquals(Math.PI * 3.5 * 3.5, circle.area(), 0.0001);
    }

    @Test
    void testCirclePerimeter() {
        Circle circle = new Circle("Circle", 3.5);
        assertEquals(2 * Math.PI * 3.5, circle.perimeter(), 0.0001);
    }
}

```

```java
package griffith;

import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

class RhombusTest {

    @Test
    void testRhombusArea() {
        Rhombus rhombus = new Rhombus("Rhombus", 4, 6, 5);
        assertEquals((4 * 6) / 2.0, rhombus.area(), 0.0001);
    }

    @Test
    void testRhombusPerimeter() {
        Rhombus rhombus = new Rhombus("Rhombus", 4, 6, 5);
        assertEquals(4 * 5, rhombus.perimeter(), 0.0001);
    }

    @Test
    void testRhombusToString() {
        Rhombus rhombus = new Rhombus("MyRhombus", 8, 10, 6);
        assertEquals("Shape: MyRhombus, Diagonals: 8.0 & 10.0, Side: 6.0", rhombus.toString());
    }
}
```

```java
package griffith;

import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

class RightAngledTriangleTest {

    @Test
    void testTriangleArea() {
        RightAngledTriangle triangle = new RightAngledTriangle("Triangle", 3, 4, 5);
        assertEquals((3 * 4) / 2.0, triangle.area(), 0.0001);
    }

    @Test
    void testTrianglePerimeter() {
        RightAngledTriangle triangle = new RightAngledTriangle("Triangle", 3, 4, 5);
        assertEquals(3 + 4 + 5, triangle.perimeter(), 0.0001);
    }

    @Test
    void testTriangleToString() {
        RightAngledTriangle triangle = new RightAngledTriangle("MyTriangle", 6, 8, 10);
        assertEquals("Shape: MyTriangle, Base: 6.0, Height: 8.0, Hypotenuse: 10.0", triangle.toString());
    }
}
```

**https://github.com/Silviu-Sri/WorksheetThree_3144757**