# Exercise 1:

## Geometry.java Class Diagram:

A screenshot of a diagram

Description automatically generated

## Relationship Descriptions:

1. Text – Shape: Shape has exactly one Text-class attribute.
2. Point – Shape: Shape has at least one point (i.e. at least one Point-class attribute)
3. Point – Colour: Point has exactly one Colour attribute
4. Shape – Rectangle: Rectangle extends Shape
5. Shape – Circle: Circle extends Shape
6. Rectangle – Prism: Prism extends Rectangle
7. Shape – Geometry: Geometry uses the Shape class

* A Geometry object uses Shapes in a TreeSet `shapes`

## Source Files Descriptions:

**Geometry.java**

The main() function instantiates the following objects: 2 Rectangles, 2 Circles, and 2 Prisms. These objects are then printed for the user to view. Geometry has a TreeSet that holds any object of class Shape or object that is of the Shape superclass. This means that this TreeSet can hold Rectangle, Circle, Prism objects.

**Prism.java**

Prism class inherits all aspects of the Shape class. Prism has its own definition for 3 of Shape's abstract classes. In addition, Prism has an attribute called height, used to calculate volume.

**Rectangle.java**

Rectangle class inherits all aspects of the Shape class. Rectangle has its own definition for 3 of Shape's abstract classes. In addition, Rectangle has attributes called length and width, used to calculate area and perimeter (volume will be 0).

**Circle.java**

Circle class inherits all aspects of the Shape class. Circle has its own definition for 3 of Shape's abstract classes. In addition, Circle has an attribute called radius, used to calculate area and perimeter (volume will be 0).

**Shape.java**

Shape is an abstract class (cannot be instantiated, but can be extended).

* Each Shape object is described by its origin and name (Point object, and Text object, respectively).
* Each Shape class has the following abstract methods (they need to be defined in Circle and Rectangle (and Prism via extension)):
* area(), perimeter(), volume()
* Each Shape class has the following concrete methods:
  + distance():

distance() can be used to calculated the distance between 2 Shapes. This method can be called via a Shape object (i.e. s1.distance(s2)) or statically called (i.e. distance(s1, s2)).

* + move():

move() moves the Shape object's origin by a specified values.

**Text.java**

Text class contains text stored as a String. Each Shape has exactly one Text object.

**Point.java**

Point class represents a point in a Shape. Each Shape has at least 1 Point.

Each Point only has one Colour and has x and y coordinates stored as type Double.

distance() can be used to calculated the distance between 2 Points. This method can be called via a Point object (i.e. p1.distance(p2)) or statically called (i.e. distance(p1, p2)).

**Colour.java**

Colour class describes a Point's colour.

# Exercise 2:

## Program Output:

A screenshot of a computer

Description automatically generatedA screenshot of a computer program

Description automatically generated

## Code:

in **607-a2-exercise-2.zip**

# Exercise 3:

## Sample Outputs:

### Sample Output 1 – Human Player vs Random Player:

A screenshot of a computer program

Description automatically generated A screenshot of a computer

Description automatically generated

### Sample Output 2 – Human Player vs Blocking Player:

### A screenshot of a computer program Description automatically generated A screenshot of a computer Description automatically generated

### A screenshot of a computer Description automatically generated

### Sample Output 3 – Human Player vs Smart Player:

## A screenshot of a computer program Description automatically generatedA screenshot of a computer Description automatically generated

## A screenshot of a computer Description automatically generated

## Code:

in **607-a2-exercise-3.zip**