COMP1006/1406 – Winter 2022

Each part of the assignment is worth 10 marks.

1 Temperature [10 marks]

Complete the provided Temperature class. Add any attributes and helper methods as needed but keep in mind that testing will involve only the methods you are asked to write/complete. You must complete the constructors and methods in the provided class (without changing any signatures, return types, or modifiers).

In this problem you will need to be able to convert temperatures between Celsius, Fahrenheit and Kelvin. For help, see https://en.wikipedia.org/wiki/Conversion_of_units_of_temperature

A Temperature object holds a single temperature value and displays it in one of the three scales. Once a scale has been set, it will display the temperature in that scale until changed. The default scale is Celsius if not specified.

The three scales are represented by Strings (class attributes) in the provided Scale class. For this assignment, the purpose of the Scale class is to provide a consistent naming scheme for the different scales. Essentially, we assign fixed names for the three scales and use these everywhere in the code.

Some examples of using a Temperature object:

```
Temperature t = new Temperature(10.1);
System.out.println(t.getScale());     // displays "CELSIUS"
System.out.println(t.toString());     // displays "10.1C"
t.setScale(Scale.FAHRENHEIT);         // change scale
System.out.println(t.toString());     // displays "50.18F"  (notice it converted the value!)
System.out.println(t.getScale());     // displays "FAHREHEIT"
t = new Temperature(12.25, "KELVIN"); // scale input is not from Scale!
System.out.println(t.getScale());     // displays "NONE"
System.out.println(t.toString());     // displays "0.0N"
```

Note: Temperature values are floating point numbers. If the expected output is "0.1F" and your output is "0.09999999999998F", that is OK. You are **not** asked to perform any string formatting in this class.

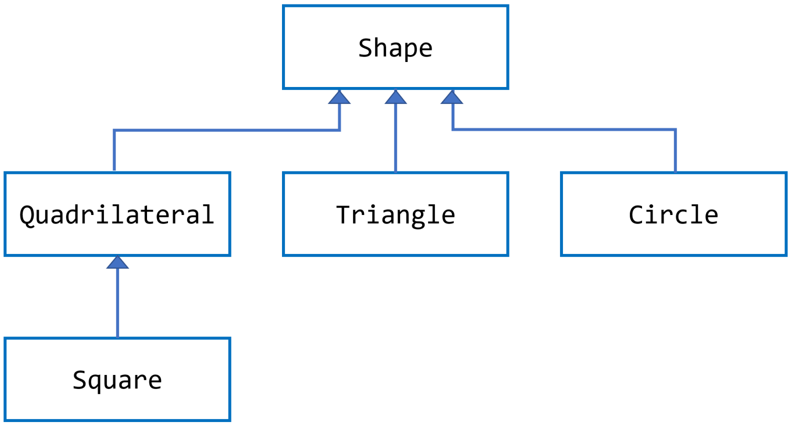
Submit your Temperature.java to Brightspace. along with your .java files.

A program called SimpleTemperatureProgram is provided with the code shown above that you can use as a starting point for your own testing if you wish.

1) You should have no static attributes or methods in your class.
2) Read the specifications in the skeleton file carefully!
(do not use Strings that look like the attributes from Scale)
3) Be sure to use **ENCAPSULATION!**
4) Remember to fill in the header comment block

2 Shapes [10 marks]

In this problem, you are provided with a Shape class and will need to create the other four classes shown in the following class hierarchy.



All shapes have an (x,y) coordinate that anchors the shape in the x-y plane. The XYCoord class is used to store (x,y) coordinates. All shapes must be able to compute their own area and perimeter[1]. You are free to add any *state* to your classes as needed. Each of the classes will have a constructor that is appropriate for the particular shape detailed as follows:

```
public Quadrilateral(XYCoord a, XYCoord b, XYCoord c, XYCoord d)
  // a quadrilateral has four straight sides and four corner points (vertices)
  // the corner points are specified by the inputs a,b,c,d
  // the sides are defined by ab, bc, cd, da
  // a is the anchor coordinate for this shape.

public Square(XYCoord anchor, double length)
  // a square is a special quadrilateral rectangle whose sides all have the same
  // length the angle at each corner is 90 degrees.
  // anchor is the bottom-left corner of the square
  // length >= 0

public Triangle(XYCoord a, XYCoord b, XYCoord c)
  // a,b,x are the three coordinates of the corners (vertices) of the triangle
  // a is the anchor coordinate for this shape

public Circle(XYCoord centre, double radius)
  // centre is the anchor coordinate for this shape
  // radius >= 0
```

You can use the provided ShapeExampleApp to help test your code. Do NOT change or submit the provided Shape.java file. Do NOT change or submit the provided XYCoord.java file. When testing, we will use the files as provided (and delete any version you submit).

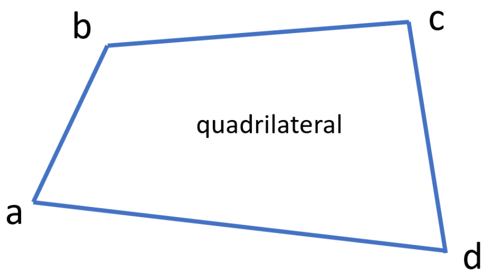
You may find the following resources helpful for your area/perimeter computations:

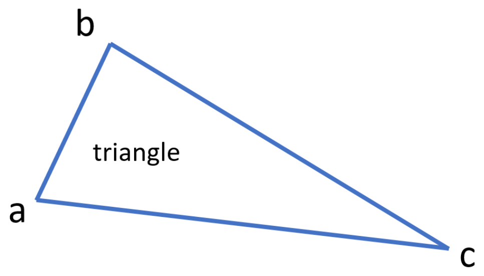
- https://byjus.com/maths/area-of-quadrilateral/
- https://byjus.com/maths/area-of-a-triangle/

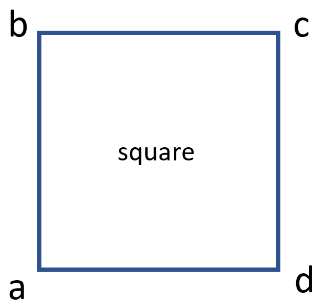
[1] The word circumference is usually used circle and shapes with curves but we will use perimeter for everything here to be consistent.

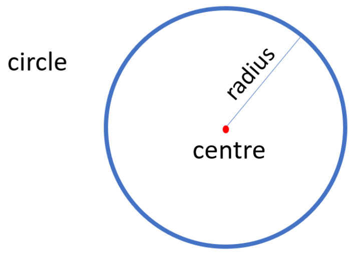
Submit your `Quadrilateral.java`, `Square.java`, `Triangle.java` and `Circle.java` files in a zip file called `a2p2.zip` to Brightspace in **A2-Part-Two**.

1) You should have no static attributes or methods in your class.
2) Read the specifications in the skeleton file carefully!
3) Be sure to use **ENCAPSULATION!**
4) Remember to create a header comment block.
5) The classes must use inheritance that follows the UML diagram!


b
c
quadrilateral
a
d


b
triangle
a
c


b
c
square
a
d


circle
radius
centre