

COM161 – Practical 10

The purpose of today's practical is to use a super-class (parent) and subclasses (child)

Scenario

You will create a superclass and three subclasses. The Shape superclass will have generic properties. These will ultimately hold the areas and boundary lengths of your shapes. The superclass will also count how many Shapes have created.

The three subclasses you need to write are called **Square**, **Circle** and **Triangle** (this will be a right-angled triangle with both the opposite and adjacent sides having the same size).

- These three subclasses should have methods which allow for the calculation of the Boundary length and for the Area of the shape.
- An application class should be developed which allow for the user to select a shape and enter in a dimension.
- An instance of the selected shape type will be used to calculate the boundary length of the shape and its area based upon the entered dimension.
- The program should print out these two pieces of information and ask the user if they wish to perform more calculations.
- The program should also output how many generic Shapes were created and how many individual Circles, Triangles, and Squares were created as well.

See the second last page of the practical for how the area and boundary lengths of the relevant shapes are calculated.

Scenario

Part One – The Super class

Create a **Shape** superclass. It will have the following

- Two instance properties – **area** and **boundary**. These will ultimately hold the areas and boundary lengths of your shapes.
- A class property called **count**. This will record how many times an instance of the **Shape** class has been created. Note that this class property is 'private'. Determine the best approach to access this property.
- Two instance methods – **getArea** and **getBoundary**. These will take in parameter that will represent the dimension entered in by the user. Both methods should return 0.

Note if this completely flummoxes you, the answer is at the back (but please attempt the super class first).

Part Two - The Subclasses

Step 1

Write a subclass called **Square**. This will extend the **Shape** class.

- Declare a class property called **countSquare**. This will record how many times an instance of the **Square** subclass has been created. Determine the appropriate access modifier for this property
- Override the two instance methods you wrote in the superclass. The first method should return the “area” of the square instance and the second should return the “boundary” of the square instance.

Step 2

Create a program that will create an instance of **Square**.

- Prompt the user to read in a dimension.
- Print out the area and boundary of the Square.
- Print out how many times an instance of the Shape class has been created
- Print out how many times an instance of the Square class has been created

Step 3

Once you have tested the **Square** subclass, create two new subclasses called **Triangle** and **Circle**. Follow the guidelines in Step 1.

Step 4

Amend your program by implementing a menu system.

- The user should be prompted for which shape they want to find the area and boundary length of.
- Once this choice has been read in, create an instance of the relevant shape and then output the area and boundary length of the shape.
- Once this is done, prompt the user to see if they want to calculate the area and boundary of another shape.
- At the end of the program output how many Shapes were created and how many individual Circles, Triangles, and Squares were created as well. See the screenshot on the following page as to how your code could look when run

Enter 1 for Square, 2 for Triangle, 3 for Circle:
Enter choice: 1
Enter dimension: 5
You choose Square
The area of the square is 25.0
The boundary of the square is 20.0
Do you want to go again? ('y' for yes): y
Enter 1 for Square, 2 for Triangle, 3 for Circle:
Enter choice: 2
Enter dimension: 5
You choose Triangle
The area of the triangle is 12.5
The boundary of the triangle is 17.071067811865476
Do you want to go again? ('y' for yes): y
Enter 1 for Square, 2 for Triangle, 3 for Circle:
Enter choice: 3
Enter dimension: 7
You choose Circle
The area of the circle is 153.93804002589985
The boundary of the circle is 43.982297150257104
Do you want to go again? ('y' for yes): y
Enter 1 for Square, 2 for Triangle, 3 for Circle:
Enter choice: 1
Enter dimension: 4
You choose Square
The area of the square is 16.0
The boundary of the square is 16.0
Do you want to go again? ('y' for yes): y
Enter 1 for Square, 2 for Triangle, 3 for Circle:
Enter choice: 2
Enter dimension: 4
You choose Triangle
The area of the triangle is 8.0
The boundary of the triangle is 13.65685424949238
Do you want to go again? ('y' for yes): n
The number of shapes created: 5
The number of squares created: 2
The number of triangles created: 2
The number of circles created: 1

Calculating boundary lengths and areas for shapes.

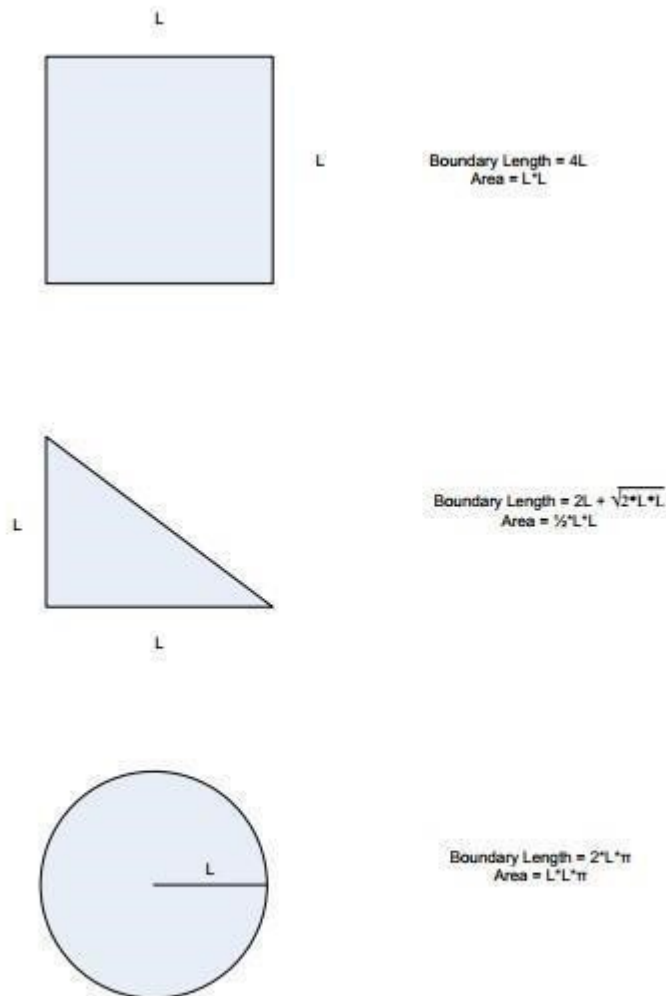


Figure 1 – Geometrical Shapes

[Hint: You can use Math.PI to draw the required constant value into your calculations. Also, you can use the Math.sqrt() method to calculate a square root.]

Shape superclass

```
4  class Shape:
5      __count = 0
6
7      def __init__(self):
8          self.area = None
9          self.boundary = None
10         Shape.__count += 1
11
12     def getArea(self, length):
13         return 0
14
15     def getBoundary(self, length):
16         return 0
17
18     @classmethod
19     def getCount(cls):
20         return Shape.__count
21
22     def printDetails(self):
23         print("Area:", self.area)
24         print("Boundary", self.boundary)
```