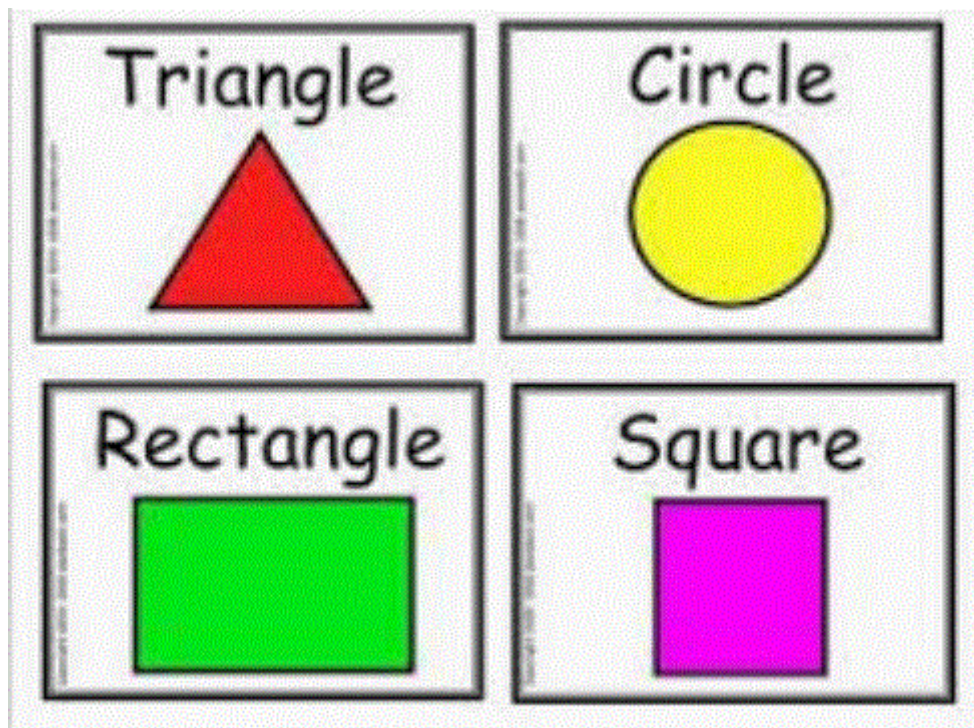


Challenge - Programming to an interface



1.

Create an **interface** ***IMyShape.java*** to represent a shape. The interface should include the method signatures

```
double calculatePerimeter();
```

```
double calculateArea();
```

```
String getShapeName();
```

2.

Develop three classes, each of which implements the ***IMyShape*** interface. The classes should represent a **circle**, **square** and **rectangle** respectively. Choose appropriate instance variables for each class and initialise these in the constructor of each.

3.

Create a class to for an *ad hoc* test of the implementations.

In the **main** method of the class randomly generate ten shapes (a mixture of circles, squares and rectangles) and store them in an array or arraylist.

Few hints :

//array to hold shapes of all three types

```
IMyShape myShape[] = new IMyShape[NUMBER_OF_SHAPES];
```

so.. this may be useful...

```
//array to hold shapes of all three types
```

```
IMyShape myShape[] = new IMyShape[NUMBER_OF_SHAPES];
```

```
//create random number generator
```

```
Random generator = new Random();
```

```
int temp = 0;
```

```
//randomly create the shapes and store in array
```

```
for (int loop=0;loop<NUMBER_OF_SHAPES;loop++){
    temp = generator.nextInt(3);
    switch (temp) {
        case 0 : myShape[loop] = new MyCircle(generator.nextDouble()*generator.nextDouble());
                break;
        case 1 : myShape[loop] = new MySquare(generator.nextDouble()*generator.nextDouble());
                break;
        case 2 : myShape[loop] = new MyRectangle(generator.nextDouble()*generator.nextDouble(),
            generator.nextInt(10),generator.nextDouble()*generator.nextDouble());
    }
}
```

4.





Also in the main method, write a loop that displays the name, area and perimeter of each shape in the array / arraylist.

Remember we are programming to the interface so ... so the output loop will look something like this...

```
for (IMyShape shape : myShape){
    System.out.print(shape.getShapeName());
    etc....
}
```

[IMyShape.java \(https://canvas.qub.ac.uk/courses/11041/files/1074280/download?wrap=1\)](https://canvas.qub.ac.uk/courses/11041/files/1074280/download?wrap=1) 

[\(https://canvas.qub.ac.uk/courses/11041/files/1074280/download?wrap=1\)](https://canvas.qub.ac.uk/courses/11041/files/1074280/download?wrap=1)

MyCircle.java (<https://canvas.qub.ac.uk/courses/11041/files/1074281/download?wrap=1>) 
(<https://canvas.qub.ac.uk/courses/11041/files/1074281/download?wrap=1>) **MyRectangle.java**
(<https://canvas.qub.ac.uk/courses/11041/files/1074285/download?wrap=1>) 
(<https://canvas.qub.ac.uk/courses/11041/files/1074285/download?wrap=1>) **MySquare.java**
(https://canvas.qub.ac.uk/api/v1/canvadoc_session?blob=%7B%22moderated_grading_whitelist%22:null,%22enable_annotations%22:null,%22enrollment_type%22:null,%22anonymous_instructor_annotations%22:null,%22submission_id%22:null,%22user_id%22:120250000000002242,%22attachment_id%22:620113,%22type%22:%22canvadoc%22%7D&hmac=b6a5cd51b16070bd2d50d9d81d869932524a0cb1) 
(https://canvas.qub.ac.uk/api/v1/canvadoc_session?blob=%7B%22moderated_grading_whitelist%22:null,%22enable_annotations%22:null,%22enrollment_type%22:null,%22anonymous_instructor_annotations%22:null,%22submission_id%22:null,%22user_id%22:120250000000002242,%22attachment_id%22:620113,%22type%22:%22canvadoc%22%7D&hmac=b6a5cd51b16070bd2d50d9d81d869932524a0cb1)
MyController.java (<https://canvas.qub.ac.uk/courses/11041/files/1074279/download?wrap=1>) 
(<https://canvas.qub.ac.uk/courses/11041/files/1074279/download?wrap=1>)