MSc Software Development

CSC7051 Programming 2

Practical 3

**Interfaces**

**Unit Testing**

**Exercise 1**

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

double calculatePerimeter();

double calculateArea();

String getShapeName();

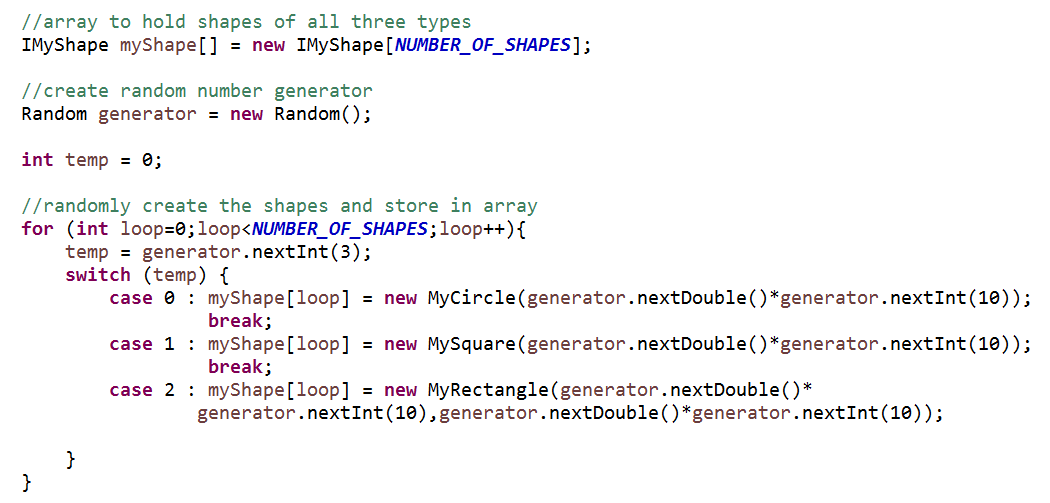
Develop three classes, each of which implements the interface. The classes should represent a **circle**, **square** and **rectangle** respectively. Choose appropriate instance variables for each class and initialise these in the constructor.

Create a class to test 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.

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

The following code will be useful when randomly generating the shapes.

import java.util.Random;



**Exercise 2**

User story – “**As a user I want to be able to pass a month value 1-12 into the system and be returned as a string the name of the month. E.g. 7 = July**”

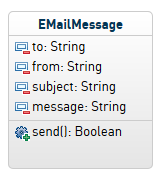
* Write a Java program to implement this functionality.
* You should use a switch statement.
* Test all possible code paths with JUnit Test cases.
* Default return (invalid number passed into the function) has been agreed to be a Number Format Exception.

**Exercise 3**

Consider the **CaesarCipher** class you developed in practical 2. Remove any *ad hoc* tests you may have created and replace with suitable **JUnit test cases**.

**Exercise 4**

1. Create the following class to represent an email message.



* There should be public getters and Setters.
* The class should include a constructor that accepts ALL the field types.
* Business rules to be implemented.

TO – limited to 30 characters. Allowable format: *text* @ *text*

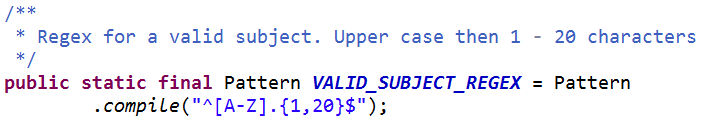
FROM – limited to 30 characters. Allowable format: text @ text

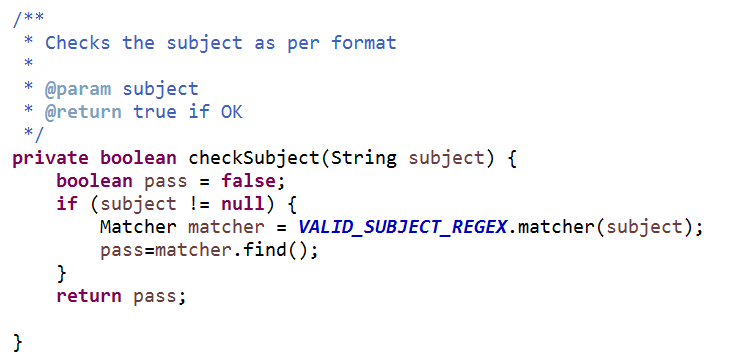
SUBJECT – 1 to 20 characters. Must begin with an upper case character.

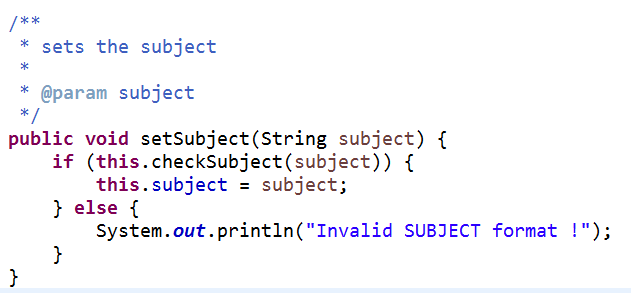
MESSAGE - 1 to 255 characters.

HINT : you can use Regex to check for the validation rules for the text fields.. for example.

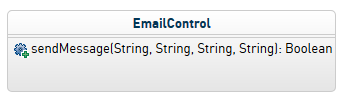








1. The **send** method will create an instance of ***EmailControl***. This class has not been developed yet and is therefore not available for you to test your class with. So you should create a **stub class** to simulate it. It should return a default **true** value.



Possible implementation of the **send** method of **EMailControl**.

EmailControl ec = new EmailControl();

ec.sendMessage(this.from, this.to, this.subject, this.message);

1. **JUnit Test** this **EMailMessage** class – including the **send** message.