Practice Assignment – 1

Save each test script and screenshots of test result as per question number. Make a zip folder of all the files. Upload the zip folder in Moodle. Source code related to each question you will find inside “Lab Practice 1” folder in Moodle. Each question is of 20 Points.

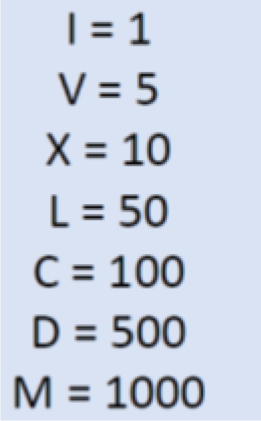
Put comment where you find bugs and fix bugs in classes and submit fixed files along with your submission.

3 things to submit for each question

- **Test Script file**
- **Screenshot of test result**
- **fixed File (if applicable, fixing bug with comment in file)**

1. Use the attached files “Circle-Radius.java” and “Circle-Radius-Test.java”. Add the code in the main function to accommodate the last test case that is when test data is "abc".

2. Consider the file “RomantoInteger.java”. This program implements a program that converts a Roman numeral that is in form of a String, to an integer. The Roman numeral system is based on seven essential numerals: I, V, X, L, C, D, and M (1, 5, 10, 50, 100, 500, and 1,000, respectively). For example



I = 1
V = 5
X = 10
L = 50
C = 100
D = 500
M = 1000

So, rules are

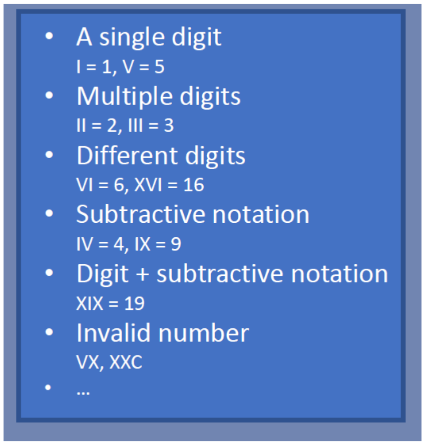
a. We can’t add more than three of the same Roman numeral together

b. If a numeral comes after a numeral that is larger or equal in value, then it must be added to the numeral before it.

c. Smaller numeral placed before a larger numeral indicates subtraction of that smaller numeral from the larger one.

d. We can’t subtract more than one value from a Roman numeral. Example of test cases for this program:

Execute these test cases using JUnit.



- A single digit
  I = 1, V = 5
- Multiple digits
  II = 2, III = 3
- Different digits
  VI = 6, XVI = 16
- Subtractive notation
  IV = 4, IX = 9
- Digit + subtractive notation
  XIX = 19
- Invalid number
  VX, XXC
- …

3. Consider Triangle.java file. Test the main method and isTriangle() method.

4. Look at the Fibonacci class. This class is an attempt at implementing the recursive method fib, which should generate the nth Fibonacci number. Create test cases for this class and run those test cases using Junit.

5. The Quadrilateral class denotes a polygon with four sides. It has two methods, isRectangle() and isSquare(). Furthermore, it also uses the classes Point, Line and Vector2D. To find if the polygon is a rectangle, vectors and dot products are used to determine if every corner forms a right angle. To find if the polygon is a square, isRectangle() is used and check if the lengths of all sides are equal.

Create test cases for each class and execute using Junit.