

Programming Assignment 2

Note: When you turn in an assignment to be graded in this class, you are making the claim that you neither gave nor received assistance on the work you turned in (except, of course, assistance from the instructor or teaching assistants).

Program: **RamString** and **RamStringTester**

Points: 100

Develop a Java class called **RamString** that correctly implements the **WackyStringInterface** that is provided. (The semantics of the methods in the interface should be obvious from their name and from the Javadoc comments in the code. If not, please ask for clarification on the Discussion Board forum.)

Important note: This project is focused on character and String manipulation and you are not allowed to use regular expressions or any classes or methods that make us of regular expressions.

Write a test driver for class **RamString** called **RamStringTester**. Specifically, you should create several different test cases for each method of in the interface. Make sure that the tests are not trivial (i.e., have a specific purpose). In other words, each test should (1) test a specific piece of functionality and (2) check that such piece of functionality behaves as expected. In addition, at least two of the tests for method **convertDigitsToRomanNumeralsInSubstring** should result in expected exceptions:

MyIndexOutOfBoundsException (provided) in one case, and **IllegalArgumentException** in the other.

Add a concise comment to each test that you implement to clarify its rationale (e.g., "This test checks whether method **convertDigitsToRomanNumeralsInSubstring** suitably throws an **IllegalArgumentException** if **startPosition** is greater than **endPosition**").

Notes:

- You cannot modify the provided interface, **WackyStringInterface**.
- You must use the provided **MyIndexOutOfBoundsException** class
- I will run your code against a standard set of test cases to make sure that you implemented the functionality of the required methods correctly.
- Comment out or remove any package statements from the files that you submit to eLearning.

Write this program in JAVA and compile it in JDK 8 or better. Follow all commenting conventions discussed in class and include a comment block at the top of each file with your name, date, the course number and section. In addition, the comment block at the top of the tester class must include a description of the program purpose, input, and output. It is expected that your program will be well documented and you are required to include a private helper method in your driver called **printHeading** that outputs the following information to the console in an easy-to-read format: your name, the project number, the course identifier, and the current semester. You will call this method as the first statement in your **main** method.

Upload the project source code files, **RamString.java** and **RamStringTester.java** to the Assignment link in eLearning (Blackboard) as two separate files (do not submit a compressed folder).

Programming Project 1 – Grading Rubric

RamString class:

Instance data member is private (5 pts.)

Constructors are implemented as specified (5 pts.)

Appropriate accessor and mutator methods included (5 pts.)

getEvenCharacters () method written as specified (10 pts.)

getOddCharacters () method written as specified (10 pts.)

countNonDigits() method written as specified (10 pts.)

isValidEmail() method written as specified (10 pts.)

ramifyString () method written as specified (10 pts.)

convertDigitsToRomanNumeralsInSubstring () method written as specified (10 pts.)

RamStringTester class:

printHeading() method included and called from main() (5 pts.)

All methods of RamString are tested (10 pt.)

Files are named and submitted as specified (5 pts.)

Appropriate use of comments (including header comments on **all files**) (5 pts.)

Total (100 pts.)

Grading Comments: