CMSC 256 – Project 2

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** implementation of the **WackyStringInterface**

Points: 100

Develop a Java class called **RamString** that correctly implements the **WackyStringInterface**. (The semantics of the methods in the interface can be determined from their name and from the Javadoc comments in the code. If not, please ask for clarification on the Discussion Board forum.) Add two constructors to your RamString class – one with a single string argument and a second, default, no-arg constructor. The no-argument constructor should set the instance variable to the string, **`"Rodney, the Ram"`** and the value of the string should never allow to be null.

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 use of regular expressions.

Write a test plan for class **RamString** using the Test Plan Template document. Specifically, you should create several different tests 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 the it 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 the method convertDigitsToWordsInSubstring 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.

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.
Upload the project source code file, **RamString.java** to Gradescope and upload your **Test Plan** to the link in Blackboard.

Programming Project 2 – Grading Rubric

RamString class:

Constructors are implemented as specified (5 pts.) __________

Mutator method and instance variable both uphold encapsulation (5 pts.) __________

getEveryThirdCharacter () method written as specified (10 pts.) __________

getEvenOrOddCharacters(String evenOrOdd) method written as specified (10 pts.) __________

countDoubleDigits () method written as specified (10 pts.) __________

isValidVCUEmail () method written as specified (10 pts.) __________

convertDigitsToRomanNumeralsInSubstring () method written as specified (15 pts.) __________

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

Programming style and readability (Including code comments) (5 pts.) __________

Test Plan:

All methods of RamString are thoroughly tested in Test Plan (20 pt.) __________

Total (100 pts.) __________