**CMSC 201 Introduction to Java**

**Project 7**

Project 7 consists of two parts:

**Part1** Create GitHub account (see attachments) and upload all java and documentation files on your GitHub account.

**Part 2** Create two java programs with JavaDoc comments

**Program 1:** Write **a JUnit test class** to test the methods length, charAt, substring, and indexOf in the java.lang.String class.

**Tips:** Use the following link and example:

[**Java API https://docs.oracle.com/javase/9/docs/api/java/lang/String.html**](file:///C:\Users\grigo\Downloads\Java%20API https:\docs.oracle.com\javase\9\docs\api\java\lang\String.html)

Example: Chapter 44, page 4, listing ArrayListTest.java (see attachment)

Name of a java class: StringTest and a file name: StringTest.java

Run Junit test

Submit a screenshot with Junit test

**Program 2:** Write **a JUnit test class** to test the method isPrime in Listing 6.7, PrimeNumberMethod.java (see attachment).

Use javadoc comments to describe a class, an interface, data fields, and methods

**Tips:**

1. Read Chapter 44.4, page 11, "Using JUnit from Eclipse."

2. Look at attached examples from Chapter 44:

PrimeNumberMethod.java, Loan.java, TestLoan.java

Name of a java class: PrimeNumberMethodTest and a file name: PrimeNumberMethodTest.java

Run Junit test

Submit a screenshot with Junit test

**Project 7 Submission requirements:**

**Deliverables:**

* Java files (source code)
* Javadoc folder with generated HTML files
* Word document should include
* Screen snapshots of outputs for program1 and 2
* Screen snapshots of your GitHub account with uploaded Project 6 files
* Lessons Learned
* Check List

**Deliverable format**: The above deliverables should be packaged in the following format.

1. FirstInitialLastName\_Project7.docx
2. FirstInitialLastName\_Project7\_Moss.zip [a compressed file containing only the following java files]

StringTest.java

PrimeNumberMethodTest.java

Your completed assignment should be submitted on **Blackboard ->Course Content-> Week xx->"Project7"** no later than the assigned due date. You should include one block comment at the top of each program containing the course name, the project number, your name, the date and platform/compiler that you used to develop the project, for example.

/\*

 \* Class: CMSC201

 \* Instructor:

 \* Description: (Give a brief description for each Program)

 \* Due: 12/13/21

 \* I pledge that I have completed the programming assignment independently.

   I have not copied the code from a student or any source.

   I have not given my code to any student.

   Print your Name here: \_\_\_\_\_\_\_\_\_\_

\*/

**Check List**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** |  | **Y/N** | **Comments** |
|  | **Source java files** |  |  |
|  | **Compressed files:** |  |  |
|  | FirstInitialLastName\_Project6\_Moss.zip |  |  |
|  | FirstInitialLastName\_Project6Completed.zip |  |  |
|  | **Program compiles** |  |  |
|  | **Program runs** |  |  |
|  | **Checklist is completed and included in the Documentation** |  |  |
|  | **Documentation file:** |  |  |
|  | **Screenshots based** |  |  |
|  | **Lessons Learned** |  |  |

**Grading Rubric**

**CMSC 201 Project 7**

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**PROGRAMMING**

Doesn’t Compile -100 pts \_\_\_\_\_ (second submission is allowed -50 pts will be deducted for 2nd attempt)

All requirements are implemented and program compiles

Accuracy:

Doesn’t Pass private instructor tests -10 pts \_\_\_\_\_

Execution: runs with errors (either run-time or logic errors) -40 pts \_\_\_\_\_

**Documentation:**

Screen shots of IDE (or console) JUnit run were not provided - 5 pts \_

Screen shots of Github account with uploaded Project 6 files were not provided - 5 pts \_

JavaDoc Comments within source code was missing or incorrect - 25 pts \_\_\_\_\_

Lessons Learned were not provided - 5 pts \_\_\_\_\_

Check List was not provided – 5pts

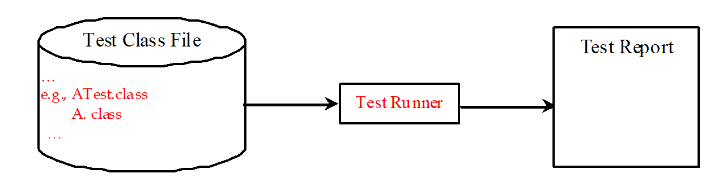
**Programming Style:**

Incorrect use of indentation, statements, structures - 5 pts \_\_\_\_\_

UI does not follow requirements - 10 pts \_\_\_\_\_

Output is easy to understand - 4 pts \_\_\_\_\_

JUnit is the de facto framework for testing Java programs. JUnit is a third-party open-source library packed in a jar file. The jar file contains a tool called test runner, which is used to run test programs. Suppose you have a class named A. To test this class, you write a test class named ATest. This test class, called a test class, contains the methods you write for testing class A. The test runner executes ATest to generate a test report, as shown in Figure below.

****