

Software Dev. & Problem Solving II

GCIS-124

Unit Testing with JUnit 5

Assignment 1.2



Goals of the Assignment

The goal of this assignment is to practice reading user input from standard input and writing unit tests using the JUnit 5 unit test framework. Please read this document **in its entirety** before seeking help from the course staff.

- **Read the assignment in its entirety.** Your questions about earlier parts of the assignment may be answered later in the description.
- The homework assignments build on the in-class activities; they are larger, more complex, and will take significantly more time to complete. Spend 20-30 minutes reviewing the lecture and in-class coding activities **before** you start the homework. If there are any activities that you cannot solve on your own, you will find the homework to be **extremely challenging**. Seek help!
- While you should **never** share your assignment solutions with other students, It is OK to share code from in-class coding activities because every student already has access to the solutions. Feel free to share your code and ask questions in the `#lecture-questions` channel on Discord!
- Work in small increments; write only a few lines of code at a time and test it to make sure that it works before writing more.
- Start your assignment early, and plan to work during times that you know that **help** will be available. This includes:
 - Your instructor's office hours (see MyCourses)
 - Virtual mentoring (see the schedule pinned in `#waiting-room` on Discord)
 - [The GCCIS Mentoring Center](#)
 - [SSE Mentoring](#) hours (10 am-6 pm, M-F)
 - Women in Computing (WiC) [Tutoring](#)
 - The RIT [Academic Success Center](#)

Activities

1. It's time for a scavenger hunt! Again! Create a folder in your Unit 1 repository named "scavenger_hunt" and add the following items:
 - a. Selfies taken at the following locations:
 - i. Your instructor's office (unless you are in an online course)
 - ii. The GCCIS Tutoring Center
 - iii. The Women in Computing (WiC) Space
 - iv. The Society of Software Engineers (SSE)
 - b. Screenshots of the following:
 - i. Your instructor's office hours
 - ii. The GCCIS Tutoring Center schedule
 - iii. The WiC Tutoring schedule
 - iv. The SSE Mentoring schedule
 - v. The Discord Virtual Mentoring Center schedule
 - vi. The email that your CA sent you to introduce themselves

Many schedules may not have been updated for the new semester yet - that's OK. It's still useful to make sure that you can find them when you need them.

2. Modify the `main` method in your `Primes` class to use a loop to prompt the user to enter a natural number, and use your `isPrime` function to print whether or not the number is prime. Stop without printing if the user enters a number that is less than 1.
3. Create a JUnit unit test class for your `Primes` class. You should write separate tests for several different values of `n` and assert that your function returns the correct result. Hint: think about edge cases, e.g. 0, 1, negative numbers, etc. Run your tests and push a screenshot of the results to your GitHub repository.
4. Modify the `main` method in your `Collatz` class to use a loop to prompt the user to enter a natural number and use your `sequence` method to print the Collatz Sequence beginning with that number. Stop without printing if the user enters a number that is less than 1.
5. Create a JUnit unit test class for your `Collatz` class. You should write separate tests for several different values of `n` and assert that your function returns the correct result. Run your tests and push a screenshot of the results to your GitHub repository.

Submission Instructions

You must ensure that your solution to this assignment is pushed to GitHub *before* the start of the next lecture period. See the [course syllabus](#) for the rubric that will be used to evaluate your submission.