Homework 4

Before attempting this project, be sure you have completed all of the reading assignments, hands-on labs, discussions, and assignments to date.

(25 points) Design a class named Clock. You should use your IDE for this exercise. The class contains private data fields for startTime and stopTime, a no argument constructor that initializes the startTime to the current time, a method named start() that resets the startTime to the given time, a stop() method that sets the endTime to the given time and a getElapsedTime() method that returns the elapsed time in seconds. Create a TestClock class to construct a Clock instance and return the elapsed time. Command line arguments should be used to send the start and end times. You should use the java.time classes. Here is sample run:

```
java TestClock 11:45:12 11:48:13
Elapsed time in seconds is: 181
```

Grading Rubric:

The following grading rubric will be used to determine your grade:

Attribute	**Exceeds**	**Meets**	**Does not meet**
Design (5 points)	(5 points) Exhibits proper use of parameters, and selection of data types all of the time. Employs correct and appropriate use of programming structures (loops, conditionals, classes etc.) all of the time. Efficient algorithms used all of the time.	(3-4 points) Exhibits proper use of parameters, and selection of data types most of the time. Employs correct and appropriate use of programming structures (loops, conditionals, classes etc.) most of the time. Efficient algorithms used most of the time.	(0-2 points) Rarely exhibits proper use of parameters, and selection of data types. Rarely employs correct and appropriate use of programming structures (loops, conditionals, classes etc.) Poorly structured and inefficient algorithms.
Functionality (10 points)	(9-10 points) Extra effort was apparent through the addition of significant and additional functionality beyond the scope of the assignment.	(7-8 points) Program fulfills most functionality. Most requirements were fulfilled. Screen captures provided	(0-6 points) Program does not fulfill functionality. Few requirements were fulfilled.

		demonstrating the successful compiling and running of the program.	
Test cases (5 points)	(5 points) Test cases provide comprehensive coverage of all code paths. Discussion of run-time errors included.	(3-4 points) Test cases provide coverage of most code paths. Test cases results well documented providing pass/fail results for each test case.	(0-2 points) No or insufficient test cases Minimal supporting evidence provided to verify testing actually took place.
Java Style Guide (5 points)	(5 points) Code impeccably neat and well-organized. Extensive In-line comments providing additional insight into code design and functionality	(3-4 points) Header comments include filename, author, date and brief purpose of the program. In-line comments used to describe major functionality of the code. Meaningful variable names and prompts applied. Class names are written in UpperCamelCase. Variable names are written in lowerCamelCase. Constant names are in written in All Capitals. Braces use K&R style**.**	(0-2 points) Code rarely follows recommended Java style guide

Submission requirements:

Your deliverables include all Java files (.java) and a single word (or PDF) document. The Java files should be named appropriately for your applications. Your word document should include screen captures showing the successful compiling and running of each application, and a detailed description of the test plan for each application. The screen captures should document your use of the IDE. The test plan should include the input, expected output, actual output and if the test case passed or failed. Submit your files to the Homework 4 assignment area no later than the due date listed in the calendar.