CSCI 4422 | CSCI 5599
HW 02 - Evolution work – improved UI

Assigned: September 09, 2019
Due: September 23, 2019 @ 23:00h

Questions (20 pts)

Download and compile the small command-line Java program Calendar.java. Calendar reads two integer inputs representing a month and a year, and prints the calendar for that month.

A test for Calendar is two integer inputs and the expected output.

1. Design at least five tests. One should be a "happy path" test, that is, a test for normal behavior. The others should use exceptional inputs. Document your tests on paper with the values and the expected results.
2. Run your tests. (You can run them by hand; we will get to automation later.) Capture and submit the results of running your tests.
3. Modify Calendar.java to include input validation. When finished, your tests related to input validation should run correctly.
4. As you make the changes, keep a simple documentation log of what you do. Note which components and methods you change and which ones you no longer use. Summarize the changes in a few words. You do not need to go to the level of which variables you create or delete or how you change the control flow. I just want the highlights. This should be no more than one page.
5. Write a short assessment of the maintainability of the software. What did the original programmers do that made it hard to change the software? What did the original programmers do that made it easy to change the software? What would you do differently if you did the assignment again? One or two paragraphs should be enough.

Submission

Submit the following items to Moodle as a **ZIP**.

1. Your five tests from step 1 above.
2. Output from your five tests on the original program.
3. Output from your five tests on your modified program.

4. The simple documentation log.
5. Your maintainability assessment.
6. Submit source code for your modified Calendar.java.

Grading

We will grade on several factors.

- (4 pts) Whether each item is included
- (3 pts) Whether the modified software works
- (4 pts) Clarity and thoughtfulness of the documentation log
- (4 pts) Clarity and thoughtfulness of the maintainability assessment
- (3 pts) Quality, maintainability, minimality of the changes you made to the code (this is why we need access to source files)
- (2 pts) Other factors to be determined while grading