CSCI 2134 Final Exam

12:00 - 3:00pm Atlantic Daylight Savings Time, Wednesday, April 15, 2020

Submitted via Git

Before the Exam:

Clone the Final Exam repository

https://git.cs.dal.ca/courses/2020-winter/csci-2134/final/????.git

where ???? is your CSID.

Be sure that that stub code compiles and that the stub Junit tests run in IntelliJ.

At the Start of the Exam

Perform a **pull** on the repository to update it. You will find that your code base has been updated and that there is much more code in them. There should be several files in the **`src`** directory and one file in the **`test`** directory (**`Tests.java`**). This is the code base for your exam.

All code modifications must take place in the java files. All written answers are to be placed in a single comment at the top of the `Tests.java` file.

Exam Rules

- The exam should be written using whatever development tools that you prefer.
- All exams will be reviewed by the course instructor
- The exam is open book. You are permitted access to
  - Your course notes
  - All code that you have written previously
  - Any books that you have at the time of writing

  You are NOT permitted to
  - Perform web searches or get help from the Internet
  - Receive assistance from any other person either locally or remotely
  - To take any other unfair advantage of the situation
- The exam is 3 hours (180 minutes) in length. The exam will open at 11:55am (ADT) on April 15, 2020 and will close at 3:05pm (ADT).
  - If you have accommodations for writing exams, these will be reflected in your exam settings.
  - If you encounter any issues, please let the course instructor know immediately either via Teams or email.

- To ask questions in the exam, email the instructor or use the private chat feature in Teams. Please do NOT post your questions in the Teams public discussion forum. The course instructor will either reply to you directly.
- The exam is out of 100 marks.

Good Luck! May the Source be with You!

Questions

1. **[10 marks]** Using Git: if you have cloned and pulled the repository for your exam, you have used git.
   NOTE: To get full marks you **MUST** perform a commit after each question. This way, if something goes wrong, I can check each part separately and give you the marks you deserve.

   Note: Your **TARGET** method will be specified at the top of the `Tests.java` file.

2. **[15 marks]** Testing
   a. Give three (3) test cases for the TARGET method (see above).
      Each test case should be one line long.
   b. Implement the unit tests in the test file.
3. **[15 marks]** Debugging: Debug the issues causing your tests to fail. There are at least a couple bugs in the code. List bugs you found and fixed. Give
   - a brief description of each bug
   - method where the bug occurs
   - how you fixed the problem
4. **[15 marks]** Defensive programming with assertions:
   a. Identify three (3) locations in the code where assertions would be appropriate. Give
      i. method where assertion should be used
      ii. what the assertions should assert
   b. Write the assertions in the code.
5. **[15 marks]** Defensive programming with exceptions
   a. Suggest one (1) exception that would be appropriate for the TARGET method and one (1) additional exception somewhere else in the code. State the condition under which the exceptions should be thrown.
   b. Implement the exceptions you suggested.
   c. Add unit tests to test that the exceptions are thrown when appropriate.
6. **[15 marks]** Procedural Refactoring
   a. 6a. Identify three (3) procedural refactoring opportunities that can be done in the code. Give
      i. a brief description of each issue
      ii. the method where the issue is
      iii. how to fix the issue
   b. Perform the refactorings on the code. Be sure to do regression testing
7. **[15 marks]** Identify three (3) class-level refactorings that can be done in the code. Give
   i. a brief description of each issue
   ii. where the issue is
   iii. what SOLID principle (if any) are violated
   iv. whether a class implementation or class interface refactoring is needed
   v. how to fix the issue