|  |
| --- |
| **CS2401 – Week 2** |

With this lab assignment, we are going to make sure that you are familiar with one IDE (IntelliJ or Eclipse) and we are going to practice testing using JUnit testing.

What is the goal of this lab?

We expect that, by the time you complete this lab you will be able to:

* Fluently use the basic functionalities of the IntelliJ or Eclipse IDE; and
* Design and run tests using JUnit testing, and report on the results.

What do you have to do? You are given three tasks, described as follow:

Task 1. You need to (install on your device and) familiarize yourself with an IDE: either IntelliJ or Eclipse. All the coding work done for this lab (and all subsequent labs this semester) should be completed using an IDE. Your compliance with this requirement will be noted (a TA will come and check your work, possibly asking you questions about how you use the IDE) and be worth 50 points of your grade on this lab.

Task 2. In a word document that you will name **YourLastName-YourFirstName-Week2.docx**, explain in 5 sentences max what testing code is about and why it is important.

Note that we will pay attention to the content (the ideas you share with us) and the form (spelling and grammar).

Task 3. You are given a java file, called Week2.java, which contains four methods that need to be tested. You have to do the following:

1/ In your word document (see Task 1 for naming of this file), clearly describe the test suite (a series of test cases) you design for each of the methods in Week2.java (one test suite per method). Each test suite should contain at least 5 test cases. Each test case has to be justified: Why did you pick this test case and not another one? Imagine you are limited by time and money about the number of test cases you can pick and run. Why would you run the test cases you propose? You have to be convincing. In particular, you have to address: WHAT each test case aims to test, and HOW you expect the method to run on this test case (what output do you expect?). Out of the 5 test cases you have to provide, at least one should be an edge case and clearly identified as such.

2/ In another java file, Week2Tester.java, write a JUnit test for each of the test cases you have described in your word file. Week2Tester.java is provided to you and each test suite already contains one test as an example. This test cannot be part of your test suite.

3/ Run your test cases and report the results in your word document. In particular, you have to report whether the method behaves as expected or not on each test case, and propose an explanation in case the method does not behave as expected.

Note: your test cases cannot include the examples given within the code.

Reminder: when designing test cases, think:

1/ regular functionality test: does the code perform as expected under normal/expected circumstances?

2/ edge case: does the code still perform when under stress of its expected conditions?

You need to have at least one edge case.

What should you turn in?

You should submit 1/ your word document and 2/ the tester file Week2Tester.java that you have completed.

How should you submit your work?

You should commit this repository to the GitHub classroom.

Failing to follow submission instructions and guidelines will result in up to 15 points off your overall grade in this lab. So please pay attention.

Additionally, your word file is expected to be neat and clear. Failing to do so will result in up to 15 points off. On the other hand, extra neat and clear work will be rewarded by up to 10 extra points.

By when should you submit your work?

Due date: Friday January 31 at 11:59pm

Lateness rule: -10 pts for 1 day of lateness / - 20 pts for 2 days of lateness / 0 after that, but you still have to turn in your work

Grading:

50 pts Task 1

20 pts Task 2

5 pts per Test suite (4 test suites, each with 5 test cases)

30 pts Task 3

5 pts per Test suite (4 test suites, each with 5 test cases)

10 pts Report of the results in word document

Grading rubrics

Task 1

50 pts Use and familiarity with IDE (Eclipse or IntelliJ)

|  |  |
| --- | --- |
| 50 pts | Student is familiarized with IDE or making significant efforts towards it |
| - 50 pts | No significant effort or no use of IDE |

Task 2

20 pts Test suite in word document

1 pt per test case / 20 test cases expected: see below for points per test case

|  |  |
| --- | --- |
| 1 pt | Per test case attempted |
| - 0.5 pt | Per test case not motivated / justified |
| - 5 pts | Grammar and spelling are poor. Sentences are not complete. |

Task 3

20 pts Test cases in main method

1 pt per test case / 20 test cases expected: see below for points per test case

|  |  |
| --- | --- |
| 1 pt | Per test case attempted |
| -0.5 pt | Per test case not consistent with description in word document |
| -0.5 pt | Incorrect syntax |

\*\* Note that with the rubric as above, students have a chance to get at least 1 point per test case if it is attempted and even if it is not perfect. They only get 0 point per test case if the test case is very wrong (i.e., two problems identified).

10 pts Report and analysis of test results

|  |  |
| --- | --- |
| 10 pts | Report of results and analysis attempted |
| -5 pts | Incomplete results |