

Introduction

Once, we asked 2110 students in the first few weeks to become familiar with Java Strings, if-statements, etc. This suggestion wasn't taken seriously, and 7 weeks later we found that many were still very shaky with these features. We don't want this to happen again, so we give this little assignment now. A2 is due *after* assignment A1 is due; we hand it out now to give you more flexibility in managing your time. We hand out A1 at the third lecture of the course.

Please keep track of the time you spend on A2. You will have to tell us the time spent when you submit A2.

Learning objectives

- Gain familiarity with String methods, if-statements, assignment, functions, and loops.
- Get used to Eclipse and JUnit testing.

Collaboration policy

You may do this assignment with one other person. If you are going to work together, then, as soon as possible —and certainly before you submit the assignment —get on the CMS for the course and do what is required to form a group. Both people must do something before the group is formed: one proposes, the other accepts.

If you do this assignment with another person, *you must work together*. It is against the rules for one person to do some programming on this assignment without the other person sitting nearby and helping. You should take turns “driving” —using the keyboard and mouse.

Academic Integrity

With the exception of your CMS-registered partner, you may not look at anyone else's code from this semester or a previous semester, in any form, or show your code to anyone else, in any form. You may not show or give your code to another person in the class.

Getting help

If you don't know where to start, if you don't understand testing, if you are lost, etc., please SEE SOMEONE IMMEDIATELY —an instructor, a TA, a consultant, the Piazza for the course. Do not wait.

What to do

File `A2.java` contains functions for you to write. Instructions are given at the top of that file and also within the body of each function, as comments. Each function you write has a “// TODO” comment. Complete the functions and submit file `A2.java` on the CMS by the end of the due date given on the CMS.

Your grade will depend only on the correctness of the functions, determined by running test cases on each function. If a function is incorrect, you may receive 0 for it. We provide JUnit testing class `A2Test.java`, with all necessary test cases. If your `A2.java` passes these test cases, you should get 100/100. We do reserve the right to deduct points even if it passes all tests if we see something fishy.

Everyone should get 100 on this assignment.

When you know your A2 is correct, *fill in the appropriate line at the top of A2 with netids and hours and minutes* and submit A2 on the CMS. Please be careful doing this in order to save us time in computing the average. Replace hh by the hours and mm by the minutes. If the minutes are 0, replace mm by 0.

CAUTION: Your two `.java` files must be in the default package. If they are not in the default package, you will lose 10 points.

Guidelines/Instructions for working with Eclipse and JUnit testing in A2

1. Create a new project `a2` (or any name you want), using menu item: `File -> New -> Java Project`.
2. In the Package Explorer, click the right arrow preceding the project name. It turns into a down arrow and you see a directory `src` (for source) with nothing in it.

3. Drag downloaded file `A2.java` on top of `src`. A window will pop up, asking whether to link or copy. Select *copy* and click `OK`.
4. There is now a right arrow before `src`. Click it. It turns into a down arrow and under it is a right arrow followed by `(default package)`. Click that right arrow. It turns into a down arrow and under it you see `A2.java`. It contains a copy of the file that you dragged.
5. Double click file name `A2.java`. It opens in the big pane to the right of the Package Explorer pane. You can now edit that file.
6. Drag downloaded file `A2Tests.java` on top of `src` and do as you did on step 3 above.
7. `A2Tests.java` does not compile because JUnit 4 is not available. The simplest way to make it available is to create a new JUnit test class (menu item `File` → `New` → `JUnit test case`). Follow directions, and when it asks whether you want JUnit 4, click `YES`. Then delete the new JUnit class.

JUNIT TESTING

We gave instructions on using Eclipse in the Eclipse tutorial on the course website. Go over that tutorial again, especially the part about JUnit testing.