CSC108H lab – week 3

This document contains the instructions for the week 4 CSC108H lab. To earn your lab mark, you must actively participate in the lab. *You don't need to finish in the time alloted, you just need to try hard.*

In this lab, you will pick a new partner, write a non-customized class, work with user input, and convert `String`s to primitive values.

1 Icebreaker and A2 partner selection

Choose a different partner than you had last week.

2 Driver and navigator

> **driver:** The person typing at the keyboard.
>
> **navigator:** The person watching for mistakes, and thinking ahead.

Throughout the lab, you'll be switching back and forth between the driver and navigator roles. The most important rule for this lab:

> **The navigator must not touch the keyboard.** If the navigator does type when they are not supposed to, the navigator will get a zero for this lab.

3 Optional activity for aspiring partners

If you plan to submit assignment one with a partner, following the instructions at

`http://www.cs.utoronto.ca/~heap/108/stgeorge.html`

for submitting `partner.txt`.

4 Starting up

Check which lab you are registered in at

`http://www.cs.utoronto.ca/~heap/108/tutorial.html`

If it is different from the lab you are currently sitting in, please make them match by either (a) moving your body to the lab you are registered in, or (b) changing the lab you are registered in to the one your body is in.

Sit down with your partner. The rest of these instructions call you two `s1` and `s2`. Pick which one is which.

`s1` should log in and start up DrJava, and be the first driver.

5 JOptionPane, Integer, Double, static reminder

As you saw in lecture, `JOptionPane.showInputDialog(String)` pops up a dialog box for user input, and then returns what the user typed. Try that now in the Interactions pane, using `"Type something"` as the argument.

You will also need these:

`Integer.parseInt(String)`, which converts the argument to an `int`, and

`Double.parseDouble(String)`, which does the same thing, only different.

6 The task

Write a class (not a customization) called `MyMath` that does simple math on values supplied by the user. Write the following `public static` method. (Put "public static" before the return type and method name.)

```
/* Prompt the user for two integers and return the sum. */
int addIntegers
```

Hint: You'll probably need to prompt for, and fetch, the integers one at a time.

Compile it, and fix any errors.

Switch roles: `s2` drives and `s1` navigates.

Test `MyMath.addIntegers` in the Interactions pane:

```
MyMath.addIntegers()   /* Should prompt for two numbers and return the sum */
```

Now write the following `public static` method.

```
/* Prompt the user for two doubles and return the sum. */
double addDoubles
```

Compile it, and fix any errors.

Switch roles: `s1` drives and `s2` navigates.

Test `MyMath.addDoubles` in the Interactions pane.

Add the following method to class `MyMath`:

```
/* Prompt the user for two doubles and return whether they are equal. */
boolean doublesEqual
```

Compile it, and fix any errors.

Switch roles: `s2` drives and `s1` navigates.

Test `MyMath.doublesEqual` in the Interactions pane.

Add the following method to class `MyMath`:

```
/* Prompt the user for a double and an int and return whether they are
   equal. 2.0 and 2, for example, are equal, but 2.1 and 2 are not. */
boolean intAndDoubleEqual
```

Switch roles: `s1` drives and `s2` navigates.

Go back and rewrite your methods (testing each one in turn) **without using any variables**.