

CSCA08H lab – week 2

This document contains the instructions for the week 2 CSCA08H lab. To earn your lab marks, you must actively participate in the lab. *You do not need to finish in the time allotted, you just need to try hard.*

Please return this handout to your TA. We use the same set of lab handouts throughout the week to save paper. We will post each handout at the end of the week.

1 Objectives

1. Make sure you have email forwarding set up properly.
2. Practice reading and using common computer science terms: *variable, method, instantiate, execute, declaration, initializing declaration, syntax, semantics* and *expression*.
3. Understand the difference between a void method and method that returns a value.
4. Practice using JFrames.

2 Setting up e-mail forwarding

Sit down with your partner. The rest of these instructions call you two `s1` and `s2`. Pick which one is which. Execute the following set of instructions twice, once for `s1` and once for `s2`.

The instructors and TAs will sometimes send important email to everyone's UTSC account. Unless this is your primary email account, you should set up "email forwarding". The instructions below give you the steps.

1. Log in: enter your user ID and password.
2. Double-click the DrJava icon. Wait while DrJava starts up; please be patient, and don't close the little window that pops up.
3. In the large right-hand pane in DrJava, which is called the *Definitions Pane*, type your favourite email address. Any mail sent to your UTSC account will be forwarded to this email address.
4. Select `File|Save`. A dialog box will appear. In the text field labelled `File Name:`, type `“.forward”`. Then select drive `G:` in the `Save In:` drop-down menu, and click the `Save` button.
5. Quit DrJava.
6. Log in to `fissure`.
7. type the following command: `“mv pc/.forward ./”`. This moves the file from the `pc` directory to your home directory. To verify, type `“more .forward”`; the contents of your `.forward` file should be displayed.

3 Using DrJava, playing with JFrames

Throughout the term, we will use the terms *driver* and *navigator*. Here are the definitions of the two roles:

driver: The person typing at the keyboard.

navigator: The person watching for mistakes, and thinking ahead.

Here is the most important rule for this lab:

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

3.1 Trying JFrames

This section is hard! Don't worry if you don't finish or don't understand everything. Remember that we will be repeating this information in lecture this week and next.

In particular, don't worry if there are several other students in your lab who finish this quickly. Grab them and make them explain it to you!

Also, if you are one of those fast students, please help the others around you.

s1 should now log in again. Throughout the lab, you'll be switching back and forth between the driver and navigator roles.

In lecture you learned about ints and doubles, variables, and how to create and manipulate JFrames. In this lab you and your partner will experiment with those concepts.

- These statements create a JFrame, make it visible, set the size, and set the title:

```
- import javax.swing.*;
- JFrame j1 = new JFrame();
- j1.show();
- j1.setSize(400, 600);
- j1.setTitle("Look! A window!");
```

- Use a couple of initializing declarations to instantiate four JFrames (call them j1, j2, j3, and j4), and practice calling the methods you saw in class on those objects. Here are (some of) those methods:

```
show, setSize, setTitle           getWidth, getHeight, getX, getY
```

- Discuss the syntactic and semantic differences between the methods in the left group and the methods in the right group. Make a *brief* list of the differences you can think of and show it to your TA.

3.2 Positioning JFrames

- Switch roles: s2 drives and s1 navigates.
- Reset the Interactions Pane.
- Use four initializing declarations to create four JFrame variables and corresponding objects, and call method show on all of them.
- Set the size of windows 1 and 3 to 100x200 pixels, and the size of windows 2 and 4 200x100 pixels. (Which JFrame method should you call?)

Does the horizontal or vertical coordinate come first in a call to setSize? Check your answer by calling getWidth and getHeight on one of the JFrames.

3.3 Working with the screen size.

Each JFrame has a method setLocation(x, y) that moves the upper left corner of the JFrame to the pixel with coordinates (x,y).

- Switch roles: s1 drives and s2 navigates.
- Using setLocation, move the first window flush against the left side of the screen, centered.
- Using setLocation, make the second window flush against the top of the screen, centered.
- The screen size is 1024x768. Figure out how to move the third window flush against the right-hand side of the screen, centered, and the fourth window flush against the bottom of the screen, centered.

3.4 Spelling with JFrames

- Switch roles: `s2` drives and `s1` navigates.
- Reset the Interactions Pane.
- Write down `s1`'s initials.
- Use JFrames to spell the initials on the screen by using a bunch of skinny and wide windows.

Show your JFrames and your Interactions Pane to your TA.