

This document contains the instructions for the week 3 CSC108H lab. To earn your lab marks, you must actively participate in the lab. *You don't need to finish in the time allotted, you just need to try hard.*

1 Objectives

- Make sure you have email forwarding set up properly.
- Write subclasses of class `JFrame`.
- Write methods and method calls.
- Experiment with typecasts.
- Solve some simple geometry problems.
- Use local variables.
- Rewrite code to eliminate local variables.
- Practice submitting files on the assignments submission page.

2 Email forwarding

Each of you should log in and look at the `.forward` file in your `H:` directory. Check the contents of this file, and make sure that the email address inside is the same as the address you set in last week's lab, in case you mistyped it. Ask your TA for help with changing it if you need it, and try emailing your UTM address to see if your mail ends up in your other mailbox. Please note that files that start with a period are *system files*, and should be handled with extreme care.

3 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 or mouse.** If the navigator does type or click when they are not supposed to, the navigator will get a zero for this lab.

4 Screen size

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.

Type the following into the Interactions pane. This gets the screen height and width and saves them in `int` variables `screenWidth` and `screenHeight`.

```
import java.awt.*;
Dimension d = Toolkit.getDefaultToolkit().getScreenSize();
int screenWidth = (int) d.getWidth();
int screenHeight = (int) d.getHeight();
d
screenWidth
screenHeight
```

Hints: Whenever you need to know how big the screen is, you can use those method calls. Remember that `(int)` is a typecast: it converts a `double` value to an `int` value. Why do we typecast? Because `d.getWidth()` and `d.getHeight()` return `doubles`, but `JFrame`'s `setSize` method needs `ints`. See what happens if, for example, you leave out the typecast in the assignment to `screenWidth`.

5 Working with the screen size

Write a subclass of `JFrame` called `MaxWindow`. `MaxWindow`'s methods are described in the instructions below. **Compile frequently:** Write the “`public class ...`” part but leave the class body empty; save (in a file called `MaxWindow.java`), compile, and fix errors. Write a method header with an empty method body; compile, and fix errors. Then write the method body, compile, and fix any errors.

Test each method from the Interactions pane right after you write it.

1. **Switch roles: s2 drives and s1 navigates.** Write a `void` method called `maximizeHeight`, which moves the window to the top of the screen and makes it as tall as the screen. Methods `setLocation` and `setSize` will come in handy. Use the `Toolkit` code from step 3 to get the screen dimensions.
2. Write a `void` method called `maximizeWidth`, which moves the window to the left of the screen and makes it as wide as the screen.
3. **Switch roles: s1 drives and s2 navigates.** Write a `void` method called `maximize`, which moves the window to the top left of the screen and makes it as big as the screen. You must do this by calling `this.maximizeHeight` and `this.maximizeWidth`.

6 Parameters and return values

Write a subclass of `JFrame` called `TilingWindow` that has the following methods. **Compile frequently.** Test each method from the Interactions pane after you write it. Save the class in a file called `TilingWindow.java`.

1. **Switch roles: s2 drives and s1 navigates.** Write an `int` method called `widthRatio` that has one parameter, a `JFrame j`, and returns the width of this window divided by `j`'s width.
2. **Switch roles: s1 drives and s2 navigates.** Write a `boolean` method called `canTileSideways` that has two parameters, an `int i` and a `JFrame j`, and returns `true` if `i` copies of `j` will fit side by side inside this window. For example, if `i` is 8, `j` is 50 pixels wide, and this window is 430 pixels wide, then the result is `true` because 8 copies of `j` can fit side by side. However, if `i` is 10, then the result is `false` because 10 copies of `j` won't fit: $10 \times 50 > 430$. You'll need to use the `Toolkit` code from step 3 to get the screen dimensions.
3. **Switch roles: s2 drives and s1 navigates.**
Rewrite `canTileSideways` so that you don't use a local `boolean` variable.

4. Write a `boolean` method called `canTile` that has two parameters, an `int i` and a `JFrame j`, and returns `true` if `i` copies of `j` will fit inside this window in a grid pattern. Hint: figure out how many times `j` fits horizontally and how many times `j` fits vertically, and go from there. Can you do this without declaring any variables?

7 More Submission Practice

Login to the submission page (<http://www.utm.utoronto.ca/submit>) Note that the submission date for **Assignment 0** has been extended to Friday. For this week's lab, you must download the file you submitted last week, add a second message to the first, and resubmit it for **Assignment 0** (this will overwrite last week's message file).

Go to last week's submission, and view the files you submitted for **Assignment 0**. Download the message file to your **H:** drive, open it up with your favorite text editor, and add a second message to the end of the first. Write something silly for this one; submit it after you get home, if you need time to think about it. When you do submit your updated `message.txt`, notice how your previous message file is overwritten by the first. Consider what this implies for your assignment submission.

Make sure that `s1` gets a chance to resubmit `message.txt` as well. For those of you who had submission problems last week and didn't get to send Steve a message, here's your chance. Make it good!