CSC108H lab – week 3

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

In this lab, you will write several subclasses of class JFrame.

1 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.

2 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.

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

Whenever you need to know how big the screen is, you can use those statements. (d.getWidth() and d.getHeight() return doubles, but JFrame’s setSize method needs ints. (int) is a typecast: it converts a double value to an int value. As an exercise, see what happens if you leave out the typecast.)
3 Working with the screen size

Write a subclass of JFrame called MaxWindow that has the following methods. Test each method from the Interactions pane after you write it. You should use the Toolkit code to get the screen dimensions. Save the class in a file called MaxWindow.java.

- **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.

- **Switch roles: s1 drives and s2 navigates.** Write a void method called maximizeWidth, which moves the window to the left of the screen and makes it as wide as the screen.

Visit the course website. Under “Assignments”, click “St. George submission instructions”. Log in using your c2xxxxxx account information. Submit MaxWindow.java for assignment A1 (you won’t be graded on the content of MaxWindow.java).

4 Parameters and return values

Write a subclass of JFrame called TilingWindow that has the following methods. Test each method from the Interactions pane after you write it. You should use the Toolkit code to get the screen dimensions. Save the class in a file called TilingWindow.java.

- **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.

- **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 × 50 > 430.

  Can you do this without declaring a boolean variable?

- **Switch roles: s2 drives and s1 navigates.** 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?

Submit TilingWindow.java through the course website for assignment A1 (the content of this file won’t be graded).