CSC108H lab – week 5

This document contains the instructions for the week 5 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 write constructors and practice `String` methods, and in the process you will learn a bit about buttons and text areas.

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 Starting up

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.

3 The task

A `javax.swing.JButton` is a clickable button. A `javax.swing.JTextArea` is a typing area. Both those things are *components*. Most GUI components go in the *content pane* of a `JFrame`.

The content pane is a *container*, because it can contain components. It has five areas: north, south, east, west, and center.

Download `BorderJFrame.java` from here:

`http://www.cs.toronto.edu/~heap/108/labs/BorderJFrame.java`

Open it in DrJava, compile it, and in the Interactions pane make a new `BorderJFrame` and `show` it. Notice that the button names don't make sense; everything is in the wrong place.

In the `BorderJFrame` constructor, figure out why and fix it. Hint: the problems happen in the calls to `c.add(...)`.

Compile it, test it, and fix any errors. You can test it with this Interactions code:

```
new BorderJFrame().show();
```

4 Another constructor

Switch roles: s2 drives and s1 navigates.

Now add a `BorderJFrame` constructor that takes 4 `String`s as parameters, and uses them for the `JButton` names in the north, south, east, and west (in that order). For example, if you type this in the Interactions pane:

```
new BorderJFrame("bn", "bs", "be", "bw").show();
```

then in the window that appears the north button will be called `bn`, the south one `bs`, and so on. In this constructor, make the text area with an empty string: `""`, rather than `"Center"`.

Compile it, test it, and fix any errors.

5 Yet another constructor

Switch roles: s1 drives and s2 navigates.

For this part, you'll need to rename the instance variables in order to keep the variable names from being confusing. Search for `northButton` and replace it with `b1`, `southButton` with `b2`, `eastButton` with `b3`, `westButton` with `b4`, and `centerArea` with `ta`.

Now add a `BorderJFrame` constructor that takes a single `String`s as a parameter. This `String` has the following format, where b1 means "button 1" (and so on), and the commas and dollar signs are separating the pieces of information:

```
b1-name,b1-loc$b2-name,b2-loc$b3-name,b3-loc$b4-name,b4-loc$text,text-loc
```

For example, with this code in the Interactions pane (exactly as shown)

```
new BorderJFrame("Danny,North$Vasiliki,Center$Hojjat,West$Wei,East$Dong,South").show()
```

then the north button would be called `Danny`, the center button `Vasiliki`, the west button `Hojjat`, the east button `Wei`, and the south text area `Dong`.

Here is a list of `String` methods that you will find useful:

- `indexOf(String s)`: return the index of s in this `String`. Use to get indices of the dollars and commas.
- `substring(int i)`: return the substring from `i` to the end. Use to extract the names and locations.
- `substring(int i, int j)`: return the substring from `i` to `j`. Use to extract the names and locations.

You must use the following process:

Try to extract the first button name and location, add it to the `BorderJFrame`, and *don't write any more code*. Instead, compile it, test it, and fix any errors. There should be only one button in the window.

Switch roles: s2 drives and s1 navigates.

Using a `substring` call in the constructor, remove the initial button information from the parameter. In our example, you should now have this:

```
Vasiliki,Center$Hojjat,West$Wei,East$Dong,South
```

Now try to extract the second button name and location, add it to the `BorderJFrame`, and *don't write any more code*. Instead, compile it, test it, and fix any errors. There should be two buttons in the window.

Repeat this process (switching roles frequently) until you have it working.