CS100J   Lab 02. Creating objects, calling their methods, and writing subclasses   Spring 2008

Name _____   Netid _____   Section time _____   Section instructor
_____

We ask you to experiment with creating objects, calling their methods, and writing subclasses. At the end of this handout is a list of oft-used methods of class JFrame.

Task 1. Practicing calling methods of objects

(a) In DrJava, using the interactions pane, create two different objects of class javax.swing.JFrame and show them. You can drag them to different places. See Sec. 1.3.2 of the text. You create a JFrame folder and store its name in a variable jf1 using the assignment "jf1= new javax.swing.JFrame();". Or, use the import statement —execute

import javax.swing.*;

jf1= new JFrame();

(b) Try out these method calls. In the table below, we show calls on the methods, using three dots ... to denote the position of an argument, which you have to fill in when you type in the call in the DrJava interactions pane. For a function call, write the value that the function call returns, so that we know that you tried such a call. For a procedure call, just put a check next to it. You can call these methods as many times as you want. Experiment.

show();	getWidth()
setSize(..., ...);	getHeight()
setLocation(..., ...);	getX()
setTitle(...);	getY()

Describe BRIEFLY, below, the syntactic and semantic differences between the methods (and calls on them) being called in the left column and those being called in the right. Hint: what kind of method is called in the left column and what kind on the right? **Syntax** refers to grammar, structure. **Semantics** refers to meaning — in programming, how a statement is executed or an expression is evaluated.

Task 2. Positioning JFrames

Reset the Interactions pane (by clicking button "reset"). Create two new JFrames (assigning their names to variables).

Make the first window 120 x220 pixels and the second window 220 x 120 pixels (by calling appropriate methods). Which coordinate comes first, the horizontal or vertical coordinate? (Write your answer below). Check your answer by calling methods getWidth and getHeight of one of the JFrames.

Drag the first window so that it is on the left of the screen and halfway down. Then, write down its x-

coordinate and y-coordinate (find these out by calling appropriate methods).

Task 3. Customizing a JFrame

1. Class Toolkit in package java.awt has a method for obtaining the screen size of your monitor (number of pixels in both dimensions). Type the following into the DrJava Interactions pane (you can copy and paste the whole thing):

import java.awt.*;
Dimension d= Toolkit.getDefaultToolkit().getScreenSize();

To the right is an object of class Dimension (in package java.awt), so you can see what it looks like. Fields height and width are of type **int**. There are methods for obtaining the width and height fields.
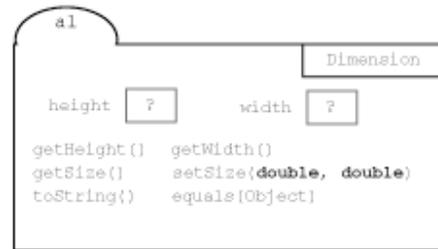
Function toString yields a String that describes the object. Try the following in the interactions pane to see this description:

d.toString()

2. In the interactions pane, type expression d (just d by itself) and hit the return key. You should then see the dimensions of your monitor. Copy them here: _____, _____
Also, type these expressions --you can copy and paste one line at a time into the Interactions pane.

        d.getHeight()
        d.getWidth()

3. Type a new class named MaxWindow into the class window (upper right pane) of DrJava. Here it is:

import javax.swing.*;
import java.awt.*;
public class MaxWindow extends JFrame {

}

Save the file with name MaxWindow.java. In the ACCEL lab, put it in a SEPARATE FOLDER. *Every time you start a new Java project, place its files in a new folder*. Now compile MaxWindow.java. Then use the Interactions pane to create and show a MaxWindow. This is just to make sure that your skeleton of the class definition is correct.

4. In class MaxWindow, type this method (but see below, first)

/** Move the JFrame to the left of the screen and make its width
    the width of the screen */
public void maximizeWidth() {
   setLocation(0, getY());
   Dimension d= Toolkit.getDefaultToolkit().getScreenSize();
   setSize((int) d.getWidth(), getHeight());
}

The above version is indented properly, but if you copy and paste you may get errors. Instead, copy the version given below, paste it into the class definition, highlight all the text in the pane, and use DrJava's indent command (in menu Edit) to indent the class properly.

```
/** Move the JFrame to the left of the screen and make its width
the width of the screen */
public void maximizeWidth() {
setLocation(0, getY());
Dimension d= Toolkit.getDefaultToolkit().getScreenSize();
setSize((int) d.getWidth(), getHeight());
}
```

5. Study method maximizeWidth, so that you know what it does and how it does it. Then, compile the class, switch to the Interactions pane, and create and show an object of class MaxWindow. Drag the window the right a bit and then call its method maximizeWidth to see whether it works properly.

6. Add to class MaxWindow a procedure maximizeHeight with the spec shown below. In writing the body of this procedure, use the body of maximizeWidth to get some ideas. This will force you to understand what maximizeWidth does. Then compile and test the procedure.

```
/** Move the JFrame to the top of the screen and make its height the height of the screen */
public void maximizeHeight()
```

7. Add to class MaxWindow a procedure with the spec shown below. Its body should simply call methods maximizeHeight and MaximizeWidth. Then compile and test it.

```
/** Move the JFrame to the left-top of the screen and make it fill the whole screen
public void maximize()
```

Task 4. Resizing a JFrame.

Create a new JDFrame named jf and show it. Try resizing the JFrame with your mouse (make it bigger). Call function isResizable() of jf to see whether JFrame jf is resizable, and put the answer here (yes or no):

Now execute the procedure call jf.setResizable(false); Try resizing jf with your mouse. Is it resizeable?

Call function isResizable() in folder jf and put the value it returns here:

Make jf resizable again.

Task 5. Find the API specifications for method setTitle of class JFrame.

(a) This task gives you practice with the general way in which one browses the API specifications. Find the root of the API specs from links subpage of the website for CS100J. Open the API specifications to class JFrame --or, simply copy and past this URL into your browser:

http://java.sun.com/javase/6/docs/api/javax/swing/JFrame.html

(b) Just read a bit to familiarize yourself with the page —spend a minute looking at it.

(c) Click the mouse near the top of the window, so that the curser appears near the top. Then search for "setTitle" --do this by typing control-F (if you are on a PC), typing setTitle into the search window, and hitting the return key. The page should scroll down to a part labeled "Methods inherited from class java.awt.Frame", and "setTitle" will be highlighted.

(d) Click on the highlighted term "setTitle". A new page will open in the browser window, which is for class Frame. Read about method setTitle. Then scroll to the top of the page and spend 1/2 minute reading about class Frame.

Task 6. Show your lab instructor or a consultant your file MaxWindow.java and this sheet.

Here are methods of class JFrame that we tend to use often:

window.show();	Show window window.
window.hide();	Hide window window.
window.getHeight()	= height of window window, in pixels
window.getWidth()	= width of window window, in pixels
window.setSize(w,h);	Set width and height of window window to w and h
window.getX()	= x-coordinate of top left corner of window window
window.getY()	= y-coordinate of top left corner of window window
window.setLocation(x,y);	Set x- and y-coordinates of top left corner of window to x, y
window.getTitle()	= the title of window window (in the title bar)
window.setTitle(s);	Set the title of window window to s (a String).
window.isResizable()	= "window window can be resized by dragging it"
window.setResizable(b);	If b is true (false) make window resizable (not resizable).