In this lab, we ask you to experiment with creating objects, calling their methods, and writing subclasses. If you are working with a partner, then one will be the driver (actually using the keyboard and mouse) and the other will be the navigator (helping the driver decide what to do, but never touching the keyboard or mouse). When moving to a new part of the assignment, please switch -- the driver becomes the navigator and the navigator becomes the driver.

Task 1. Practicing calling methods of objects

(a) In DrJava, create two different JFrames and show them. You can drag them to different places. (Remember that you need to import the classes of package javax.swing. See Sec. 1.4.2 of the text. You create a JFrame folder and store its name in a variable jf1 using the assignment "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 tries 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.

<table>
<thead>
<tr>
<th>show();</th>
<th>getWidth()</th>
</tr>
</thead>
<tbody>
<tr>
<td>setSize(..., ...);</td>
<td>getHeight()</td>
</tr>
<tr>
<td>setLocation(..., ...);</td>
<td>getX()</td>
</tr>
<tr>
<td>setTitle(...);</td>
<td>getY()</td>
</tr>
</tbody>
</table>

Describe BRIEFLY, below, the syntactic and semantics 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?

Task 2. Positioning JFrames

Reset the Interactions pane. Create two new JFrames (assigning their names to variables).

Make the first window 120 x 220 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 this into the DrJava Interactions pane (you can copy and paste the whole thing):

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

Here's what an object of class Dimension (in package java.awt) looks like:

![Dimension Class](image)

Fields height and width are of type **int**.

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.

```java
d.getHeight()
d.getWidth()
```

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

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

Save the file with name MaxWindow.java, compile it, and 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)

```java
/** 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.

```java
/** 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 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. Then compile and test the procedure.

/** Move the JFrame to the topof 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()