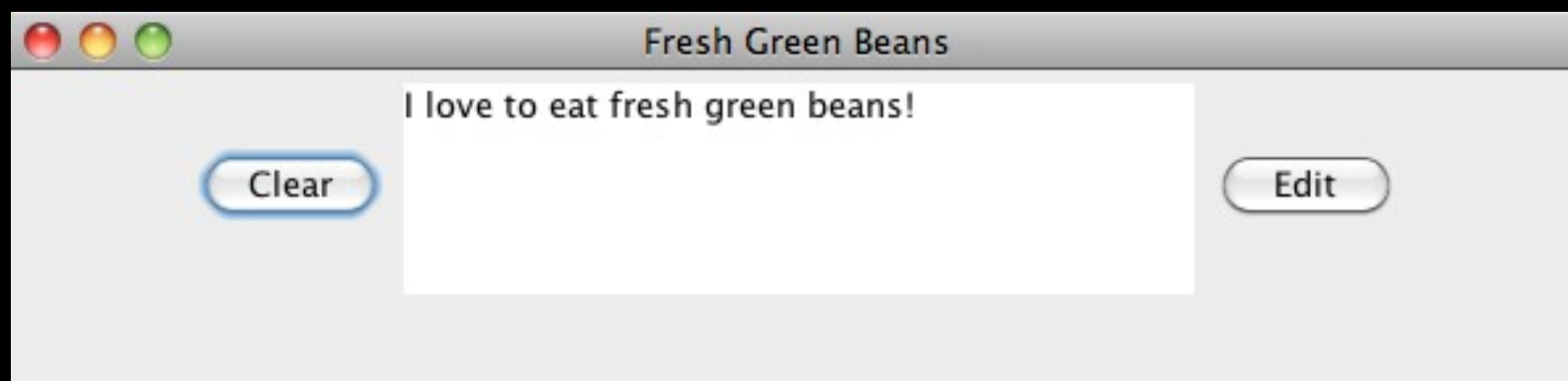


# Slides for the Week

CS273 Laboratory 12

This week's lab focuses  
on documentation

You will write code from scratch to create a small interface that looks like this:



using the JFrame class.

**This lab is different from  
the others in 2 ways.**

First, there is **no** starter  
code. You will be writing  
your program from **scratch**.

**This lab is different from  
the others in 2 ways.**

Second, instead of relying on  
the lab handout to provide  
step-by-step information on  
how to write the code....

javax.swing

## Class JFrame

[java.lang.Object](#)

└ [java.awt.Component](#)

└ [java.awt.Container](#)

└ [java.awt.Window](#)

└ [java.awt.Frame](#)

└ **javax.swing.JFrame**

### All Implemented Interfaces:

[ImageObserver](#), [MenuContainer](#), [Serializable](#), [Accessible](#), [RootPaneContainer](#), [WindowConstants](#)

```
public class JFrame
```

```
extends Frame
```

```
implements WindowConstants, Accessible, RootPaneContainer
```

An extended version of `java.awt.Frame` that adds support for the JFC/Swing component architecture. You can find task-oriented section [How to Make Frames](#).

The `JFrame` class is slightly incompatible with `Frame`. Like all other JFC/Swing top-level containers, a `JFrame` contains a `JRootPane` pane should, as a rule, contain all the child-managed components. This is different from the AWT `Frame` class, which has been overridden to forward to the container as necessary. All means that you can write

```
frame.add(child);
```

And the child will be added to the content pane. The content pane will always be non-null. Attempting to set it to null will cause an `IllegalArgumentException`. The content pane will always have a `BorderLayout` manager set on it. Refer to [RootPaneContainer](#) for details on adding, removing and setting the `LayoutManager`.

Unlike the AWT `Frame`, `JFrame` does not have a default close operation. The default close operation is set by the `setDefaultCloseOperation()` method. The default close operation is `WindowConstants.DO_DEFAULT_CLOSE_OPERATION`. The `setDefaultCloseOperation()` method can be used to set the default close operation to `WindowConstants.DO_NOthing_ON_CLOSE`.

Some of the recent packages in the Java 2 SDK, Version 1.4.2, contain the `JFrame` class. In a multi-screen environment, you can create a `JFrame` on a different screen device. See [Frame](#) for more information.

...you'll be figuring it out  
yourself using the same  
documentation that industrial-  
strength Java programmers use.



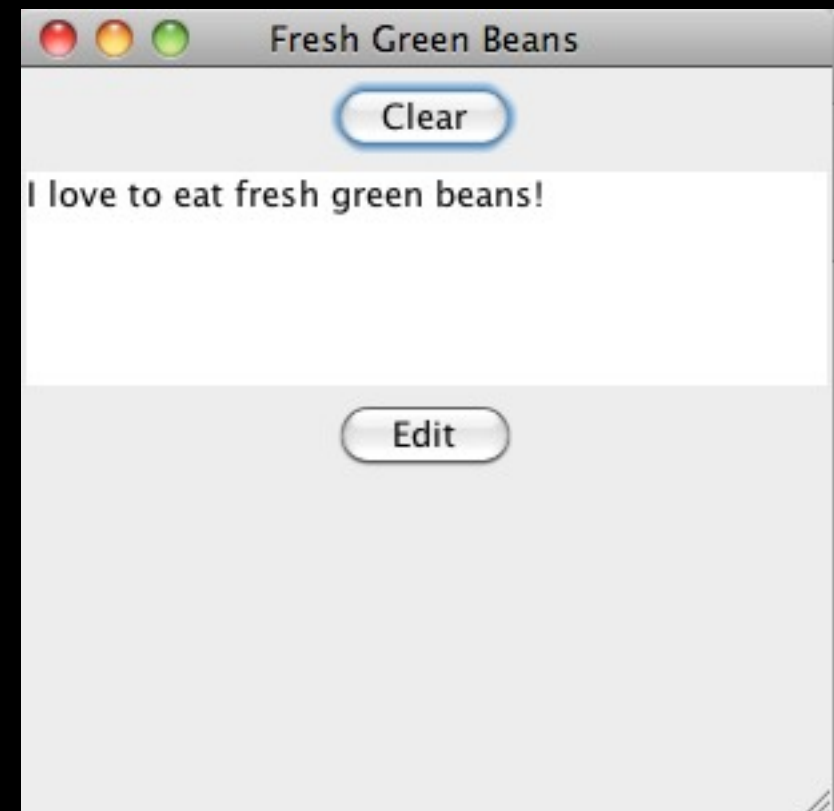
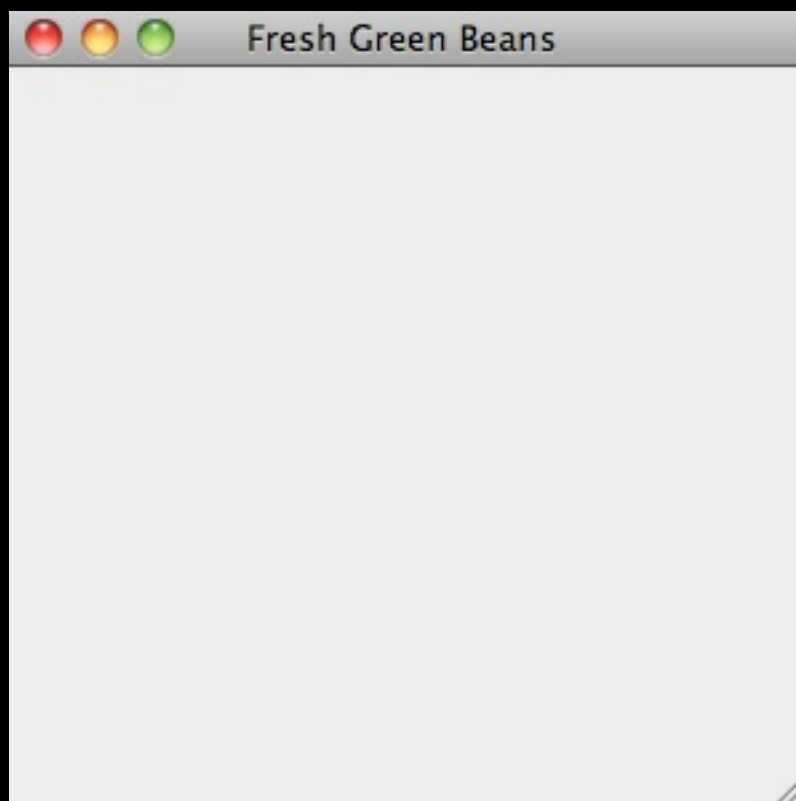
Available at:

[http://docs.oracle.com/javase/6/  
docs/api/](http://docs.oracle.com/javase/6/docs/api/)

One thing to keep in mind:

When you are running your code, you may need to  
resize your JFrame window so that the components on  
your interface get drawn.

Your mileage may vary.



**Good Luck!**

If you have any questions the TAs and I are happy to help.