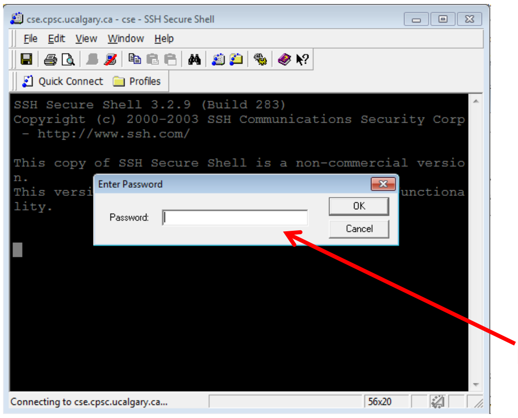
Graphical User Interfaces: Part II

Dialog Boxes

- Typically take the form of a small window that 'pops up' during program execution.


cse.cpsc.ucalgary.ca - cse - SSH Secure Shell
File Edit View Window Help
Quick Connect
Profiles
Enter Password
Password:
OK
Cancel
Connecting to cse.cpsc.ucalgary.ca...
56x20

```
SSH Secure Shell 3.2.9 (Build 283)
Copyright (c) 2000-2003 SSH Communications Security Corp
 - http://www.ssh.com/

This copy of SSH Secure Shell is a non-commercial versio
n.
This versi                                      unctiona
lity.
```

Part of the login 'dialog'

`JDialog` Example

- Location of the full example:

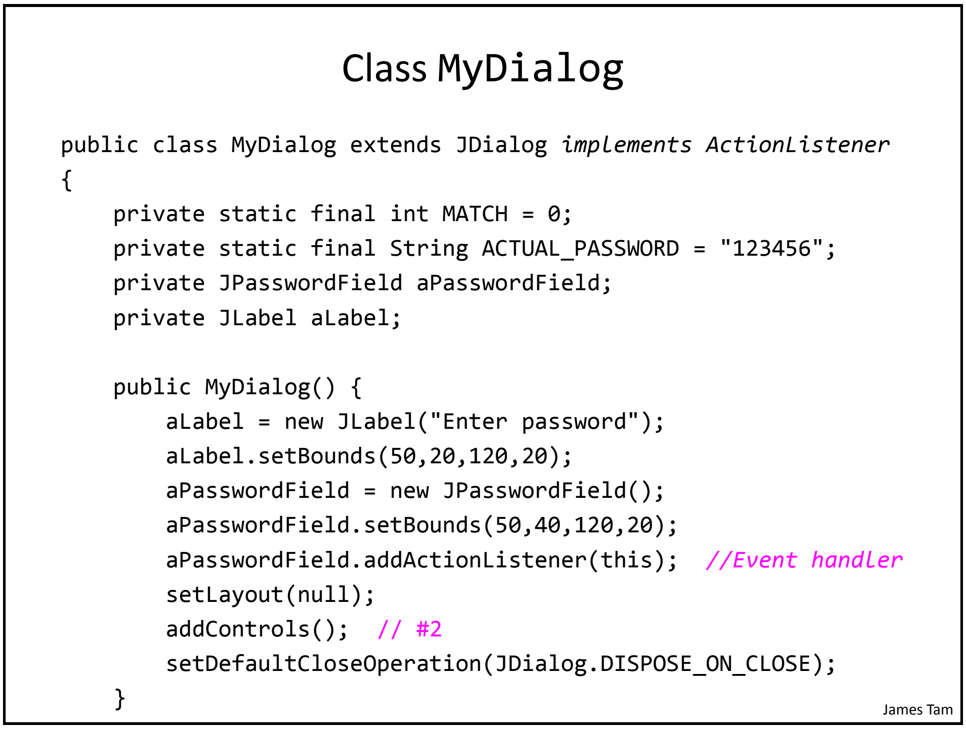
  `/home/233/examples/gui/9dialogExample`

James Tam

The `Driver` Class

```
public class Driver
{
    public static void main(String [] args)
    {
        MyDialog aDialog = new MyDialog();
        aDialog.setBounds(100,100,300,200);
        aDialog.setVisible(true);
    }
}
```

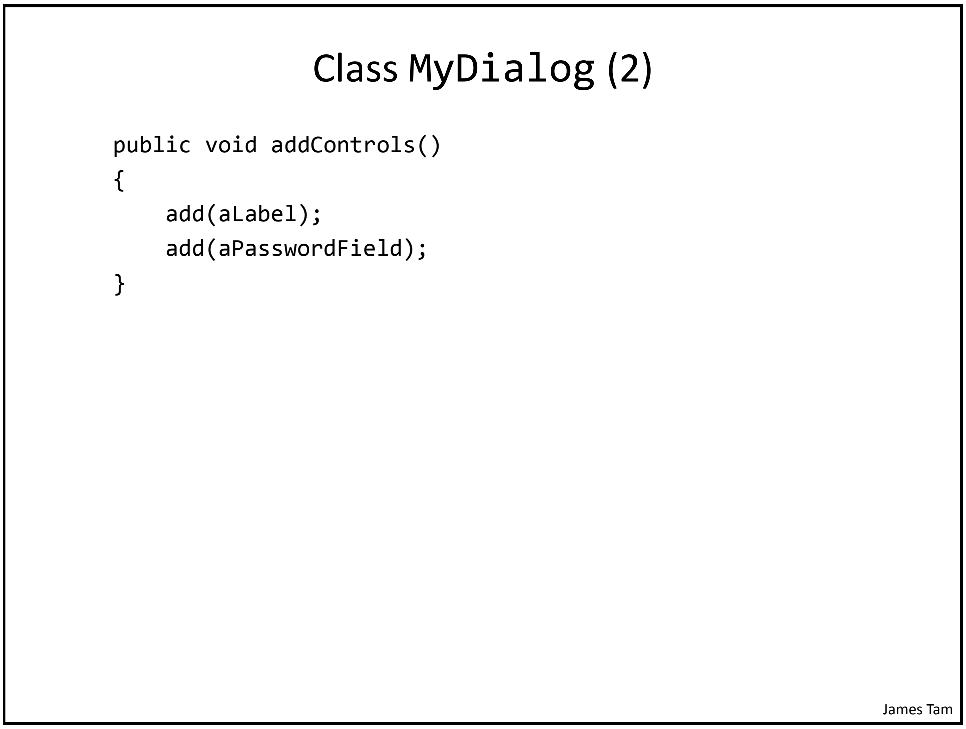
James Tam



Class MyDialog

```
public class MyDialog extends JDialog implements ActionListener
{
    private static final int MATCH = 0;
    private static final String ACTUAL_PASSWORD = "123456";
    private JPasswordField aPasswordField;
    private JLabel aLabel;

    public MyDialog() {
        aLabel = new JLabel("Enter password");
        aLabel.setBounds(50,20,120,20);
        aPasswordField = new JPasswordField();
        aPasswordField.setBounds(50,40,120,20);
        aPasswordField.addActionListener(this);  //Event handler
        setLayout(null);
        addControls();  // #2
        setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
    }
```

James Tam



Class MyDialog (2)

```
    public void addControls()
    {
        add(aLabel);
        add(aPasswordField);
    }
```

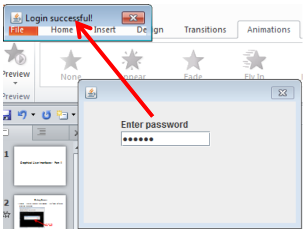
James Tam

Class MyDialog (3)

```
public void actionPerformed(ActionEvent e) {
    Component aComponent = (Component) e.getSource();
    if (aComponent instanceof JPasswordField) {
        JPasswordField aPasswordField =
          (JPasswordField) aComponent;
        String passWordEntered = new
          String(aPasswordField.getPassword());
        if (passWordEntered.compareTo(ACTUAL_PASSWORD)
                                              == MATCH)
            loginSuccess();  // #4
        else
            loginFailed()
    }
}
```

Class MyDialog (4)

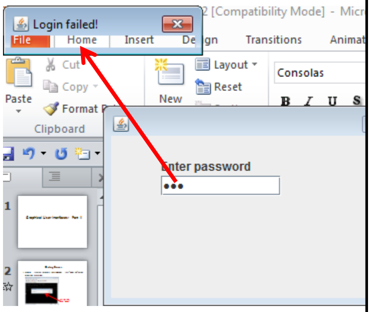
```
public void loginSuccess() {
    JDialog success = new JDialog();
    success.setTitle("Login successful!");
    success.setSize(200,50);
    success.setVisible(true);
    cleanUp(success);
}
```



```
public void cleanUp(JDialog popup) {
    try
        Thread.sleep(3000);
    catch (InterruptedException ex)
        System.out.println("Program interrupted");
    this.setVisible(false);
    this.dispose();
    popup.setVisible(false);
    popup.dispose();
    System.exit(0);  // Dialog cannot end whole program
}
```

Class MyDialog (5)

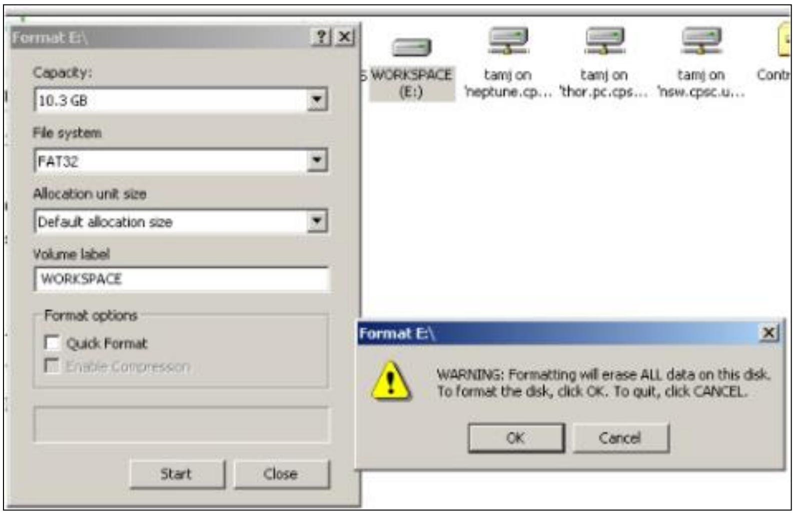
```
public void loginFailed()
{
   JDialog failed = new JDialog();
    failed.setTitle("Login failed!");
    failed.setSize(200,50);
    failed.setVisible(true);
    cleanUp(failed);
}
public void cleanUp(JDialog popup) {
    try
        Thread.sleep(3000);
    catch (InterruptedException ex)
        System.out.println("Program interrupted");
    this.setVisible(false);
    this.dispose();
    popup.setVisible(false);
    popup.dispose();
    System.exit(0);  // Dialog cannot end whole program
}
```



Enter password
•••

Dialog Boxes And "User-Friendly Design"

- Note: used *sparingly* dialog boxes can communicate important information or to prevent unintentional and undesired actions.



Format E:\

Capacity:
10.3 GB

File system
FAT32

Allocation unit size
Default allocation size

Volume label
WORKSPACE

Format options
☐ Quick Format
☐ Enable Compression

Start   Close

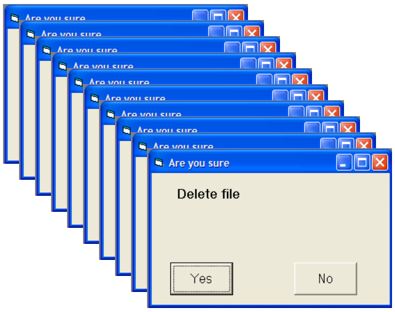
Format E:\

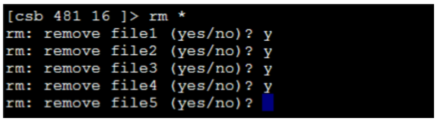
WARNING: Formatting will erase ALL data on this disk.
To format the disk, click OK. To quit, click CANCEL.

OK   Cancel

Dialog Boxes And “User-Friendly Design” (2)

- They interupt the regular use of the program so make sure they are only used sparingly
  - …they can easily be over/misused!)

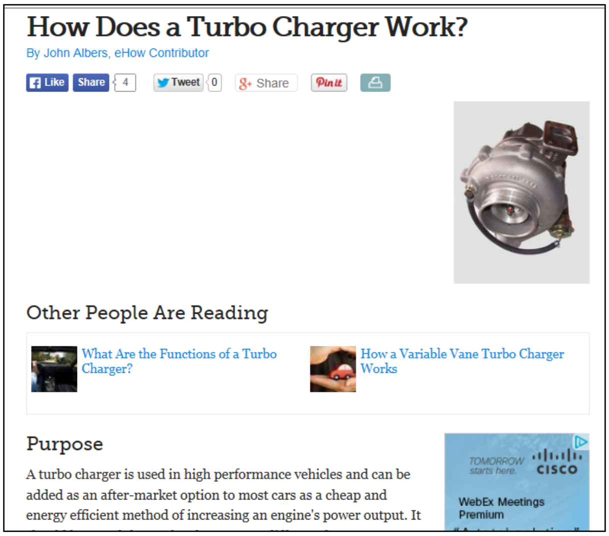




```
[csb 481 16 ]> rm *
rm: remove file1 (yes/no)? y
rm: remove file2 (yes/no)? y
rm: remove file3 (yes/no)? y
rm: remove file4 (yes/no)? y
rm: remove file5 (yes/no)?
```

Dialogs Are Frequently Used Online

- Great! I’ve got the info that I need.



How Does a Turbo Charger Work?

By John Albers, eHow Contributor

Other People Are Reading

What Are the Functions of a Turbo Charger?

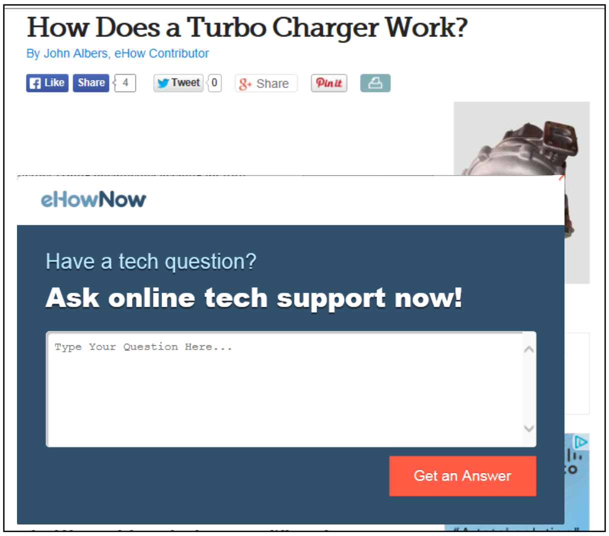
How a Variable Vane Turbo Charger Works

Purpose

A turbo charger is used in high performance vehicles and can be added as an after-market option to most cars as a cheap and energy efficient method of increasing an engine's power output. It

Dialogs Are Frequently Used Online

- Hey I was reading that!



How Does a Turbo Charger Work?

By John Albers, eHow Contributor

eHowNow

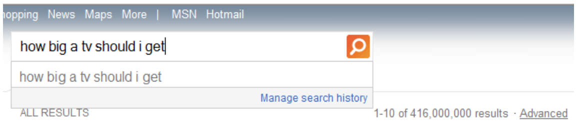
Have a tech question?

Ask online tech support now!

Type Your Question Here...

Types Of Input Text Fields: Short

- `JTextField` (you already learned): Used to get short user input
  - e.g., entering login or personal information.



how big a tv should i get

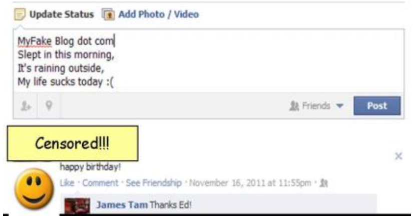
how big a tv should i get

1-10 of 416,000,000 results · Advanced

Bing search query

Types Of Input Text Fields: Long

- Getting more extensive input
  - e.g., feedback form, user review/comments on a website
  - Requires the use of another control: `JTextArea`



MyFake Blog dot com
Slept in this morning,
It's raining outside,
My life sucks today :(

happy birthday!

Like · Comment · See Friendship · November 16, 2011 at 11:55pm

James Tam Thanks Ed!

Facebook status update field

James Tam

The `Driver` Class

```
public class Driver {
    public static void main(String [] args) {
        JFrame frame = new JFrame();
        frame.setSize(400,250);
        JTextArea text = new JTextArea();
        JScrollPane scrollPane = new JScrollPane(text);
        text.setFont(new Font("Times",Font.BOLD, 32));
        for (int i = 0;i < 10; i++)
            text.append("foo" + i + "\n");
        frame.add(scrollPane);
        MyDocumentListener l = new MyDocumentListener();
        (text.getDocument()).addDocumentListener(l);
        frame.setVisible(true);
        frame.setLayout(null);
        frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    }
}
```

James Tam

The Text Listener: MyDocumentListener

```
public class MyDocumentListener implements DocumentListener {
    public void changedUpdate(DocumentEvent e) { // Modify
       System.out.println("updated");
       method(e);
    }

    public void insertUpdate(DocumentEvent e) { // Add
       System.out.println("insert");
       System.out.println(e.getLength());
       method(e);
    }
    public void removeUpdate(DocumentEvent e) { // Remove
       System.out.println("removed");
       method(e);
    }
}
```

James Tam

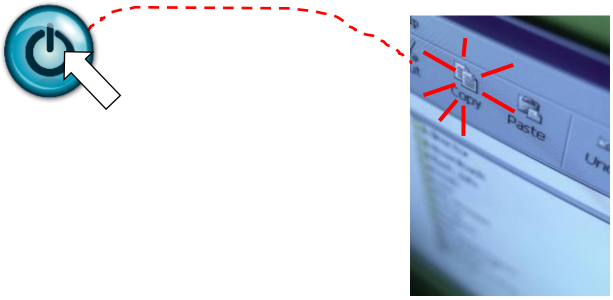
The Text Listener: MyDocumentListener (2)

```
    public void method(DocumentEvent e) {
       Document d = e.getDocument();
       try {
           String s = d.getText(0,d.getLength());
           System.out.println(s);
       }
       catch (BadLocationException ex)
       {
           System.out.println(ex);
       }
    }
```

James Tam

Controls Affecting Other Controls

- As previously shown this is not an uncommon occurrence

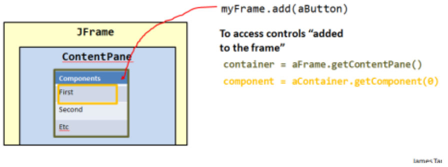


- The code to react to the event allows for easy access to the control that raised the event.

James Tam

Ways Of Accessing Other Controls

1. Via Java Swing containment
   - Example to illustrate with `JButton` control:
   - `/home/233/examples/gui/6controlAffectControls`



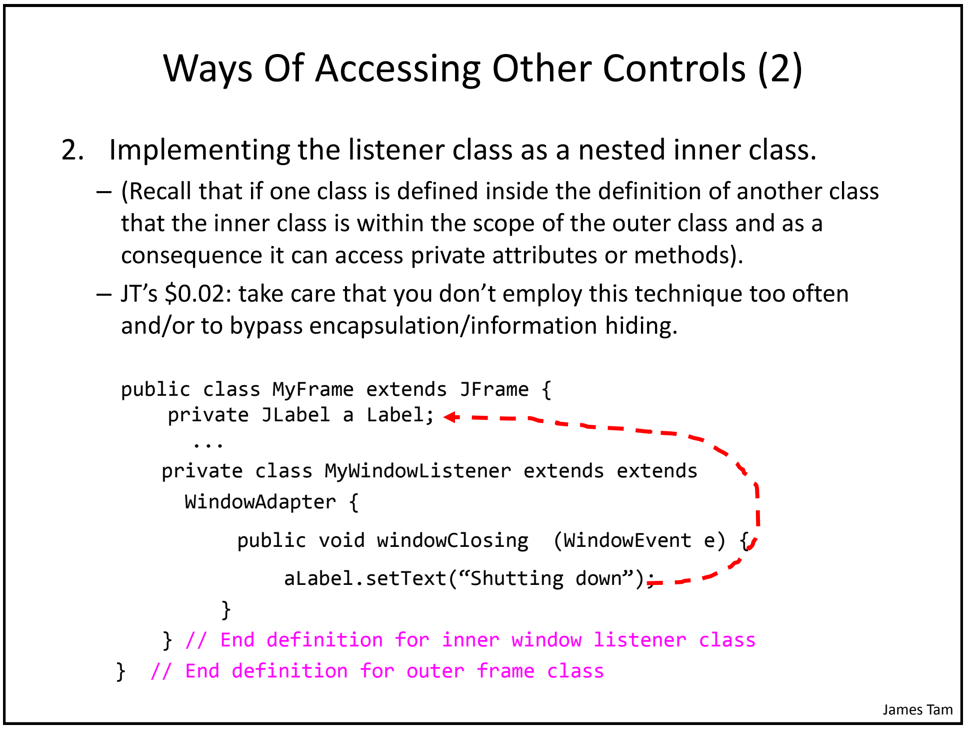
```
myFrame.add(aButton)
```

To access controls “added to the frame”

```
container = aFrame.getContentPane()
component = aContainer.getComponent(0)
```

   - JT’s $0.02
     - Stylistically acceptable (of course!)
     - Can be challenging to track down specific container/method

James Tam

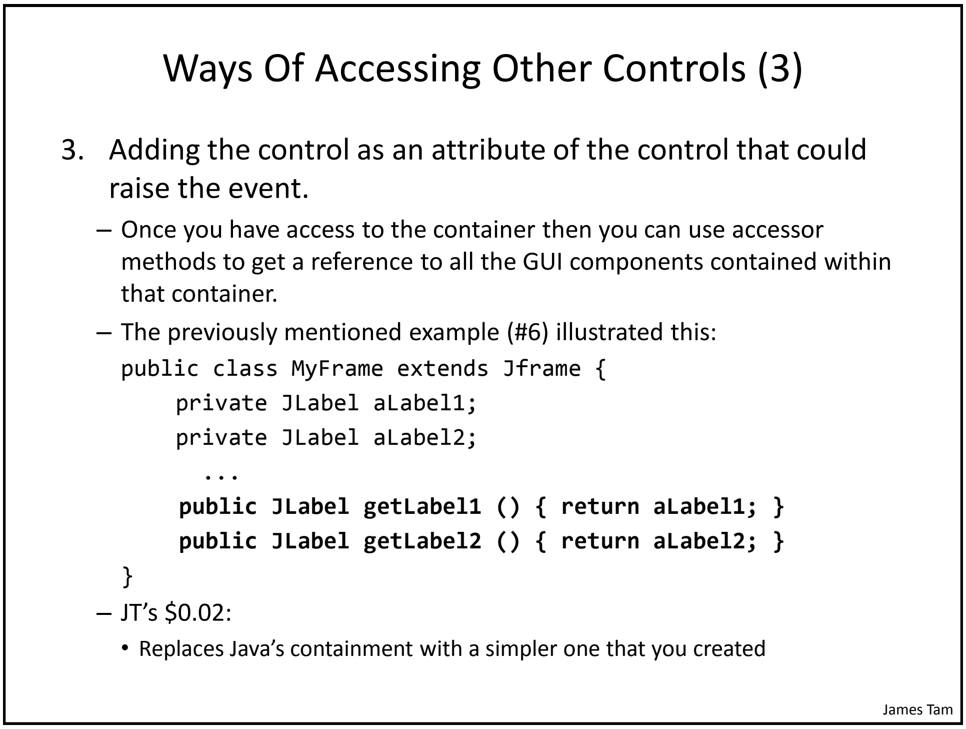


Ways Of Accessing Other Controls (2)

2. Implementing the listener class as a nested inner class.
   - (Recall that if one class is defined inside the definition of another class that the inner class is within the scope of the outer class and as a consequence it can access private attributes or methods).
   - JT's $0.02: take care that you don't employ this technique too often and/or to bypass encapsulation/information hiding.

```
public class MyFrame extends JFrame {
    private JLabel a Label;
      ...
   private class MyWindowListener extends extends
     WindowAdapter {
         public void windowClosing  (WindowEvent e) {
             aLabel.setText("Shutting down");
        }
   } // End definition for inner window listener class
}  // End definition for outer frame class
```

James Tam



Ways Of Accessing Other Controls (3)

3. Adding the control as an attribute of the control that could raise the event.
   - Once you have access to the container then you can use accessor methods to get a reference to all the GUI components contained within that container.
   - The previously mentioned example (#6) illustrated this:

```
public class MyFrame extends Jframe {
     private JLabel aLabel1;
     private JLabel aLabel2;
       ...
     public JLabel getLabel1 () { return aLabel1; }
     public JLabel getLabel2 () { return aLabel2; }
}
```

   - JT's $0.02:
     - Replaces Java's containment with a simpler one that you created

James Tam

Ways Of Accessing Other Controls (4)

– Note: adding one control as an attribute of another control need not be limited only to actual 'containers' such as `JFrame` or `JDialog`

– Example (button event changes a label)

```
public class MyButton extends JButton {
     private JLabel aLabel;
     ...
     public Jlabel getLabel() {  return(aLabel); }
}

public class MyButtonListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        MyButton aButton = (MyButton) e.getSource();
        JLabel aLabel = aButton.getLabel();
    }
}
```

James Tam

Example Illustrating The Third Approach[1] And Adding Graphics To Controls

- Location of the complete example:

```
/home/233/examples/gui/11containment
```

[1] Adding a control as an attribute of another control need not be limited only to traditional container classes such as a `JFrame`

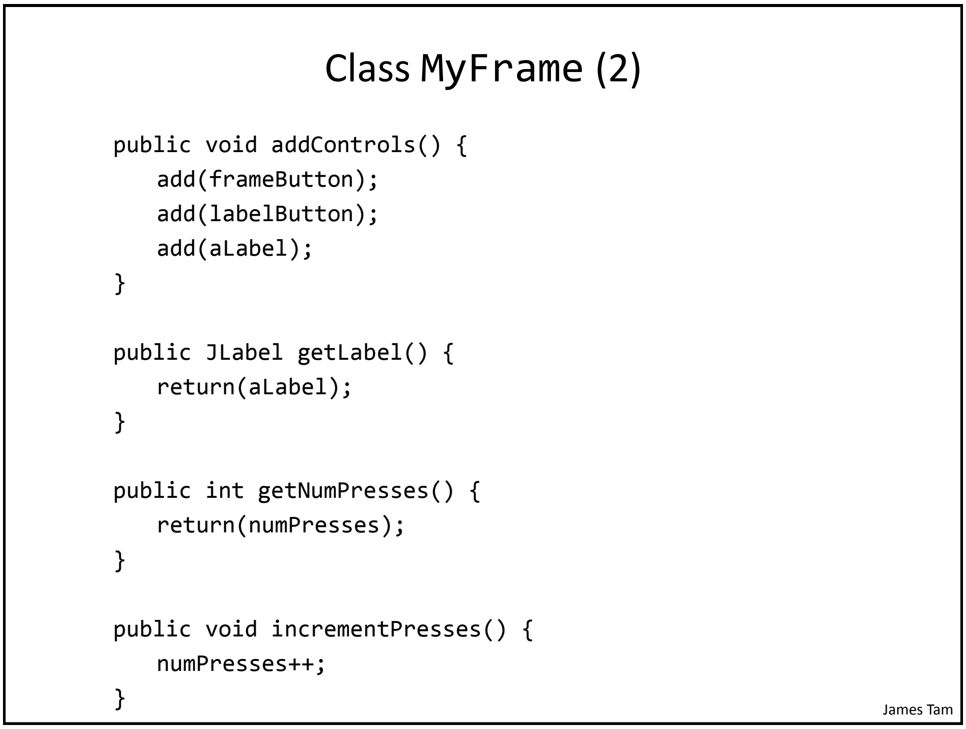
James Tam

The Driver Class

```
public class Driver
{
    public static void main(String [] args)
    {
       MyFrame aFrame = new MyFrame();
       aFrame.setVisible(true);
    }
}
```

Class MyFrame

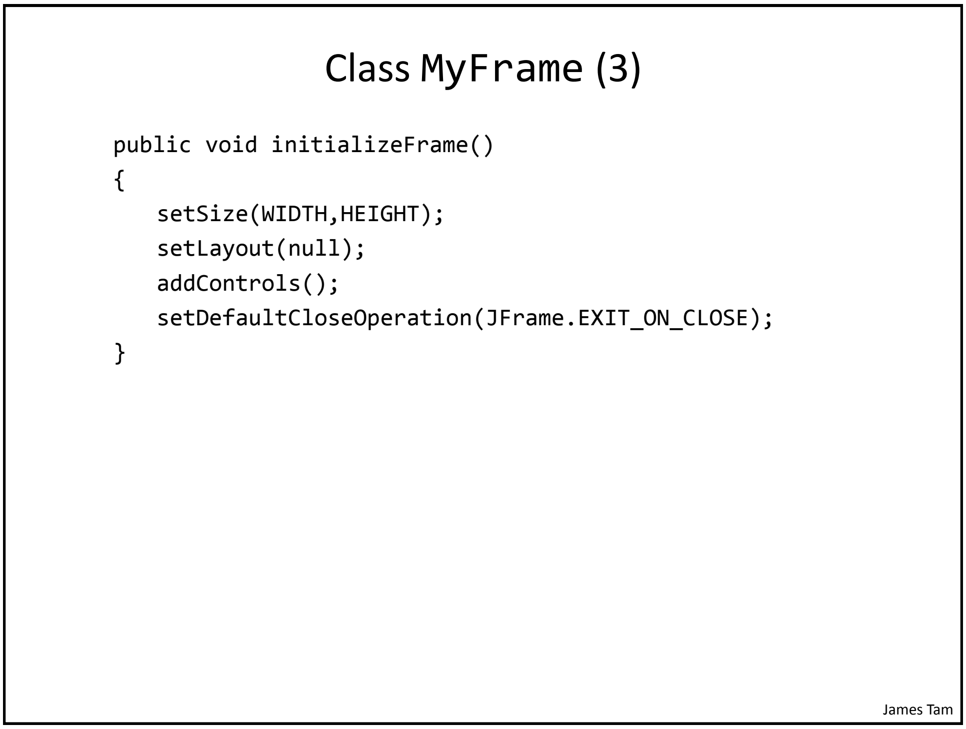
```
public class MyFrame extends JFrame
{
    public static final String DEFAULT_LABEL_STRING = "Number
      presses: ";
    public static final int WIDTH = 700;
    public static final int HEIGHT = 300;
    private MyButton frameButton;
    private MyButton labelButton;
    private JLabel aLabel;
    private int numPresses;

    public MyFrame()
    {
       numPresses = 0;
       initializeControls();
       initializeFrame();
    }
```



Class MyFrame (2)

```
public void addControls() {
   add(frameButton);
   add(labelButton);
   add(aLabel);
}

public JLabel getLabel() {
   return(aLabel);
}

public int getNumPresses() {
   return(numPresses);
}

public void incrementPresses() {
   numPresses++;
}
```

James Tam



Class MyFrame (3)

```
public void initializeFrame()
{
   setSize(WIDTH,HEIGHT);
   setLayout(null);
   addControls();
   setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
```

James Tam

Class MyFrame (4)

```
    public void initializeControls() {
        ImageIcon anIcon = new ImageIcon("IconP
        frameButton = new MyButton("Affects
                                   window",anIcon,this);
        frameButton.setBounds(50,100,150,20);
        frameButton.addActionListener
          (new FrameButtonListener()); // Frame events only
        labelButton = new MyButton("Affects label",anIcon,this);
        labelButton.setBounds(250,100,150,20);
        labelButton.addActionListener
          (new LabelButtonListener());  // Label events only
        aLabel = new JLabel(DEFAULT_LABEL_STRING +
                            Integer.toString(numPresses));
        aLabel.setBounds(450,100,150,20);
    }
}
```

No path provided: location is the same directory as the program

James Tam

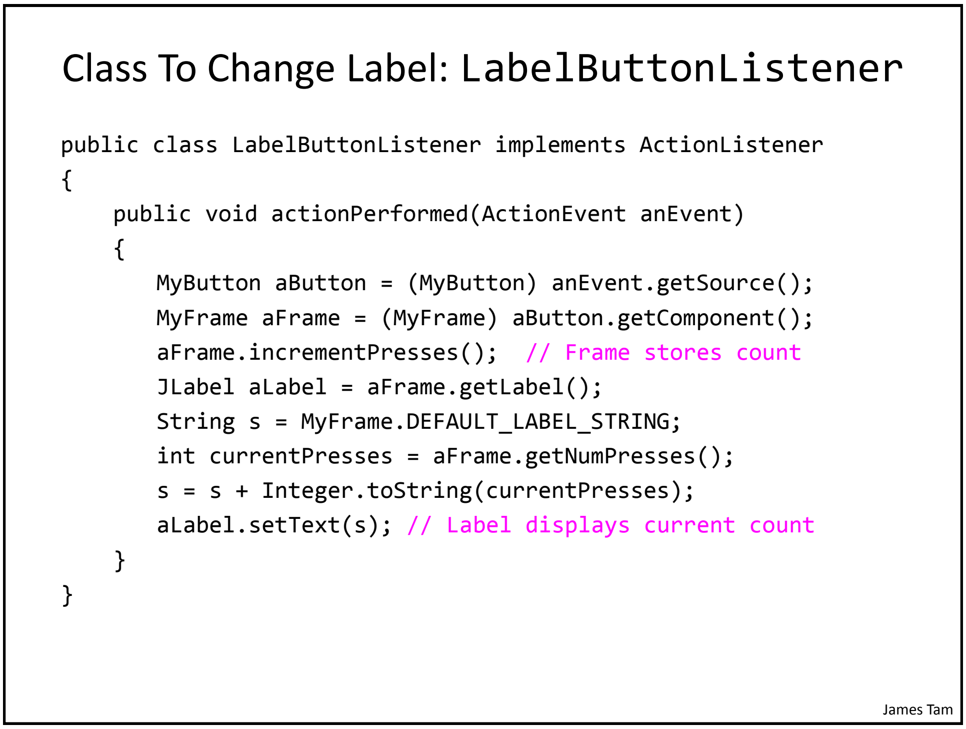
Class MyButton

```
public class MyButton extends JButton
{
    private Component aComponent;

    public MyButton(String s,
                    ImageIcon pic,
                    Component aComponent)
    {
        super(s,pic);
        this.aComponent = aComponent;
    }

    public Component getComponent()
    {
        return(aComponent);
    }
}
```

Each instance will have a reference to a Java GUI widget (label, frame etc.)

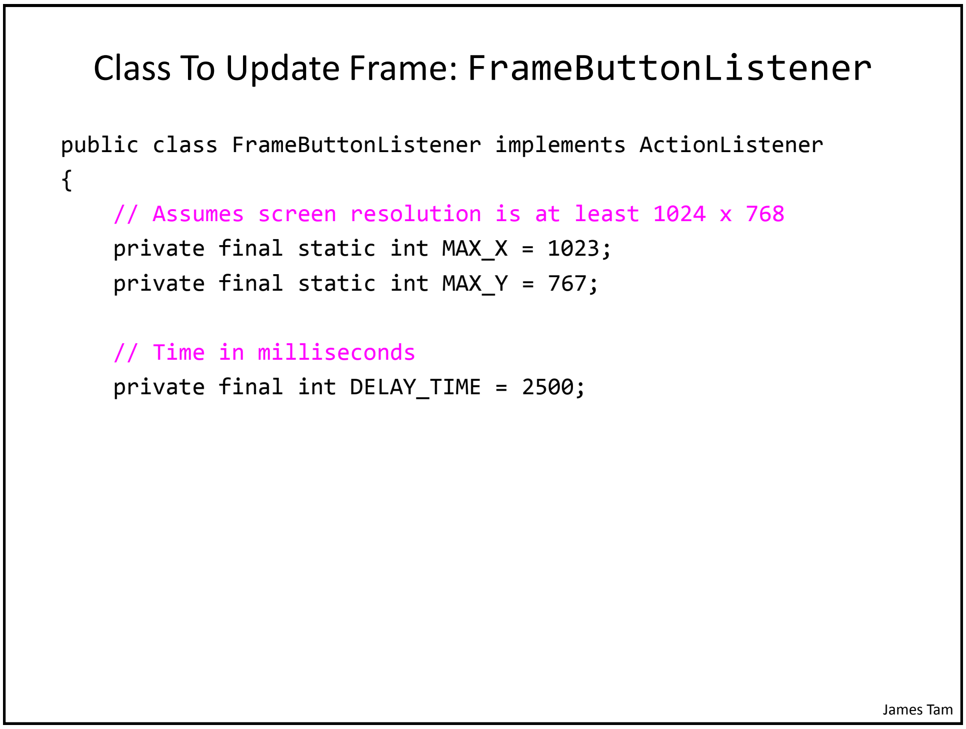
Image reference passed onto the appropriate super class constructor

James Tam



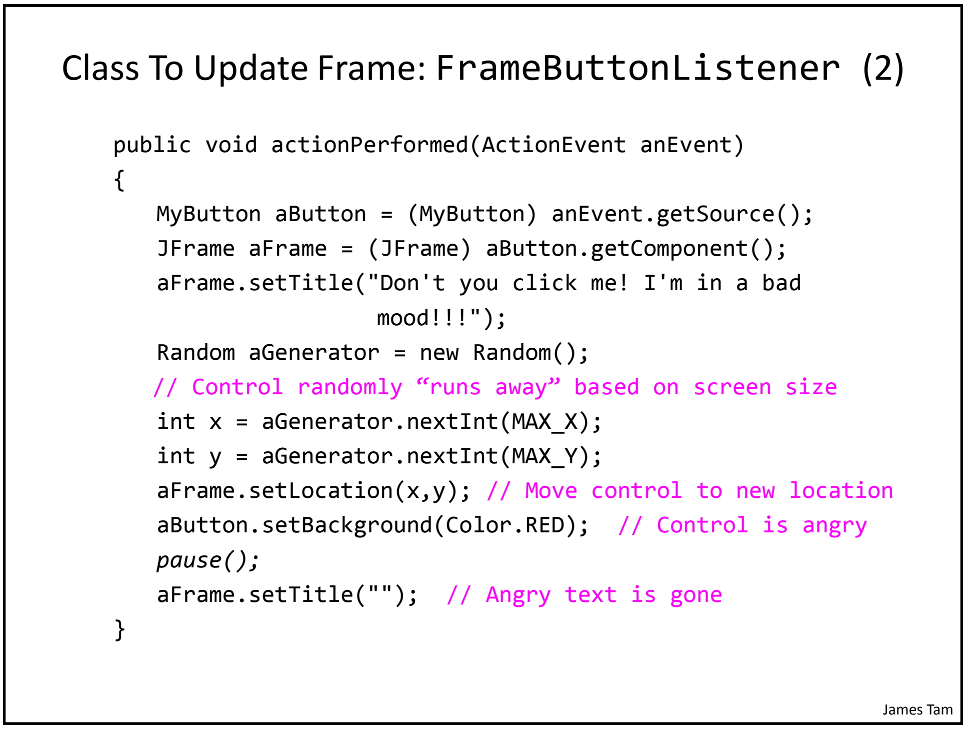
Class To Change Label: `LabelButtonListener`

```
public class LabelButtonListener implements ActionListener
{
    public void actionPerformed(ActionEvent anEvent)
    {
       MyButton aButton = (MyButton) anEvent.getSource();
       MyFrame aFrame = (MyFrame) aButton.getComponent();
       aFrame.incrementPresses();  // Frame stores count
       JLabel aLabel = aFrame.getLabel();
       String s = MyFrame.DEFAULT_LABEL_STRING;
       int currentPresses = aFrame.getNumPresses();
       s = s + Integer.toString(currentPresses);
       aLabel.setText(s); // Label displays current count
    }
}
```



Class To Update Frame: `FrameButtonListener`

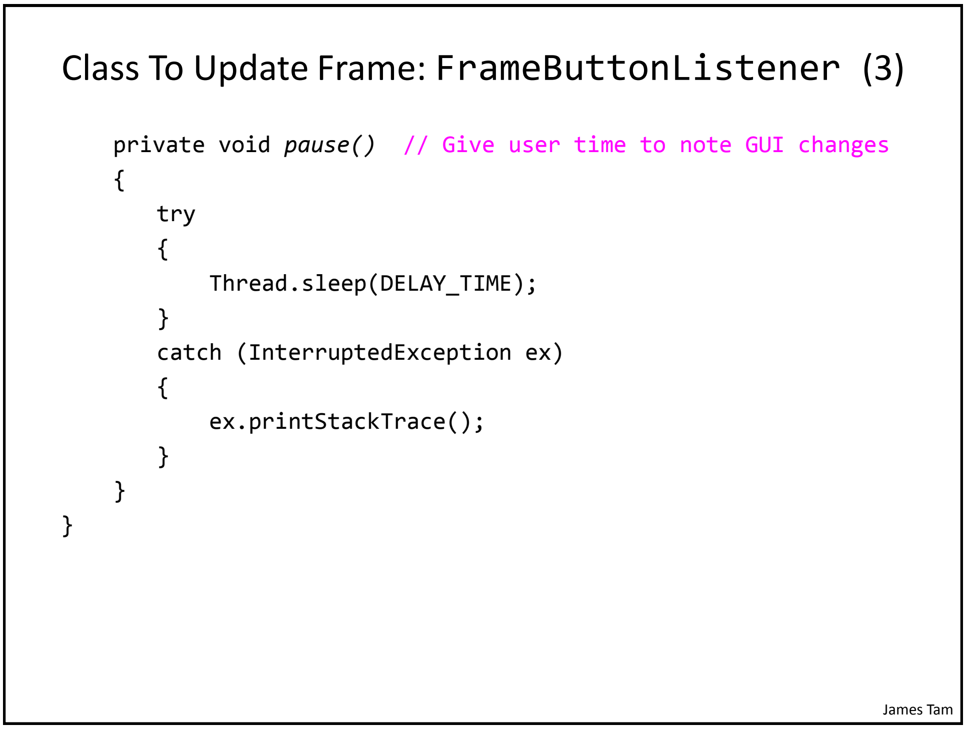
```
public class FrameButtonListener implements ActionListener
{
    // Assumes screen resolution is at least 1024 x 768
    private final static int MAX_X = 1023;
    private final static int MAX_Y = 767;

    // Time in milliseconds
    private final int DELAY_TIME = 2500;
```



Class To Update Frame: FrameButtonListener (2)

```
    public void actionPerformed(ActionEvent anEvent)
    {
       MyButton aButton = (MyButton) anEvent.getSource();
       JFrame aFrame = (JFrame) aButton.getComponent();
       aFrame.setTitle("Don't you click me! I'm in a bad
                        mood!!!");
       Random aGenerator = new Random();
       // Control randomly "runs away" based on screen size
       int x = aGenerator.nextInt(MAX_X);
       int y = aGenerator.nextInt(MAX_Y);
       aFrame.setLocation(x,y); // Move control to new location
       aButton.setBackground(Color.RED);  // Control is angry
       pause();
       aFrame.setTitle("");  // Angry text is gone
    }
```

James Tam



Class To Update Frame: FrameButtonListener (3)

```
    private void pause()  // Give user time to note GUI changes
    {
       try
       {
           Thread.sleep(DELAY_TIME);
       }
       catch (InterruptedException ex)
       {
           ex.printStackTrace();
       }
    }
}
```

James Tam