
Debugging

Appendix A - Lecture Slides

Errors

- When we have errors in our code, we need to *debug* our code to find and fix those errors
- Types of errors
 - -
 -
 - -
 -
 - -
 -

Problem Solving

- Debugging : finding and fixing errors in code
- `System.out.println` prints to the console window
- use print statements to see where you are:
 - `System.out.println("a");`
 - `System.out.println("b");`
- use print statements to print out values as you go

```
public void init( )
{
    System.out.println( "A" );
    setLayout( new FlowLayout( ) );
    System.out.println( "B" );
    setupList( );
    System.out.println( "C" );
    textarea = new JTextArea( 5,10 );
    add(textarea);
    System.out.println( "D" );
    addListItemsToTextArea( );
    System.out.println( "E" );
}
```

Problem Solving

- Use print statements to print out values as you go
- If 'C', 'D', and 'E' don't print, then there's a problem in the method `setupList()`

```
public void init( )
{
    System.out.println( "A" );
    setLayout( new FlowLayout( ) );
    System.out.println( "B" );
    setupList( );
    System.out.println( "C" );
    textarea = new JTextArea( 5,10 );
    add(textarea);
    System.out.println( "D" );
    addListItemsToTextArea( );
    System.out.println( "E" );
}
```

Printing Values

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class DebugValue extends JApplet
    implements ItemListener
{
    double cat = 4;
    double dog = 10;
    double horse = 350;
    double costOfItem = 0;
    double quantity;
    JComboBox list;
    JTextField qty;
    JLabel price;
    public void init( )
    {
        setLayout( new FlowLayout( ) );
        qty = new JTextField( 4 );
        list = new JComboBox( );
        price = new JLabel( "0.0" );
        list.addItem( "cat" );
        list.addItem( "dog" );
        list.addItem( "horse" );
        list.addItemListener( this );
        add( qty );
        add( list );
        add( price );
    }
}
```

```
public void itemStateChanged( ItemEvent ie )
{
    Object src = ie.getSource( );
    if( ie.getStateChange( ) == ItemEvent.SELECTED )
    {
        if( src == list )
        {
            quantity = Integer.parseInt( qty.getText( ) );
            double totalCost = costOfItem * quantity;
            price.setText( "" + totalCost );
        }
    }
}
```



Printing Values

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class DebugValue extends JApplet
    implements ItemListener
{
    double cat = 4;
    double dog = 10;
    double horse = 350;
    double costOfItem = 0;
    double quantity;
    JComboBox list;
    JTextField qty;
    JLabel price;
    public void init( )
    {
        setLayout( new FlowLayout( ) );
        qty = new JTextField( 4 );
        list = new JComboBox( );
        price = new JLabel( "0.0" );
        list.addItem( "cat" );
        list.addItem( "dog" );
        list.addItem( "horse" );
        list.addItemListener( this );
        add( qty );
        add( list );
        add( price );
    }
    public void itemStateChanged( ItemEvent ie )
    {
        Object src = ie.getSource( );
        if( ie.getStateChange( )==ItemEvent.SELECTED)
        {
            if( src == list )
            {
                quantity = Integer.parseInt( qty.getText( ) );
                System.out.println( "qty=" + quantity );
                System.out.println( "cost="+costOfItem);
                double totalCost = costOfItem * quantity;
                System.out.println( "total=" + totalCost );
                price.setText( "" + totalCost );
            }
        }
    }
}
```

OUTPUT:
qty=3.0
cost=0.0
total=0.0



Printing Values - fix

```
public void itemStateChanged( ItemEvent ie )
{
    Object src = ie.getSource( );
    if( ie.getStateChange( )==ItemEvent.SELECTED)
    {
        if( src == list )
        {
            quantity = Integer.parseInt( qty.getText( ) );
            String sel = (String )list.getSelectedItemAt( );
            if ( sel.equals( "dog" ) )
                costOfItem = dog;
            else if ( sel.equals( "cat" ) )
                costOfItem = cat;
            else if ( sel.equals( "horse" ) )
                costOfItem = horse;
            double totalCost = costOfItem * quantity;
            price.setText( "" + totalCost );
        }
    }
}
```



Debugging Layouts

- Change the background color of your panels and components so you can see how they stretch.
- Try calling `setPreferredSize(Dimension)` on your components

Debuggers

- Debuggers are programs that help you determine what is going on in your code.
- Use it to:
 - Print out values of variables
 - Create break points to stop execution and verify variables
 - Step through code line by line
 - And lots more
- Graphical IDE's such as Eclipse have debuggers you can use.

Summary

- **Methods for Debugging**
 - Print out markers throughout code
 - Print out values of variables
 - Color-code panels
 - Use a debugger