PRG 421

Java Programming II

Rosa Williams

Facilitator

Class Topics

- Week One: User Interfaces
- Week Two: GUI Components
- Review → Week Three: Files
- Week Four: Applets and Graphics
- Week Five: Database Connectivity and Mobile Computing

Week Three: Files			
	Details	Due	Points
Learning Team Algorithm	Create an algorithm for your Week 5 program. The algorithm can be either in the form of a flow chart or pseudo code. Submit the algorithm to your facilitator.	Assignment Tab Tuesday, December 10, 2013 6:00 PM	5

Week Three: Files			
	Details	Due	Points
Individual Income Tax Program	Individual Assignment: This assignment should be an application, not an applet, written in Java with a graphical user interface. Write an application that contains a JButton, and three JTextFields. This program should calculate Income Tax. The user enters gross pay and total deductions in two of the text fields. When the user presses the button, the income tax due should be displayed in the third text field. To calculate the income tax, first subtract the deductions from the gross pay. This difference is the net pay. If the net pay is less than 10 thousand dollars, then the income tax due is zero. If the net pay is greater than or equal to 10 thousand dollars, then the tax due is twenty five percent of the net pay. All input and output should be through the GUI not via the console. Submit the .java source code files with the correct file names for all classes used. The program must consist of at least one public class. Programs are expected to compile and run/execute correctly. There should be proper documentation (comments) in the source code. The documentation should include a block containing the name of the program and the name of the student, and a short description of the program.	Assignment Tab Tuesday, December 10, 2013 6:00 PM	15

Questions?

Class Topics

- Week One: User Interfaces
- Week Two: GUI Components
- Week Three: Files
- Week Four: Applets and Graphics
- Week Five: Database Connectivity and Mobile Computing

Week Four: Applets and Graphics			
	Details	Due	Points
Objectives	1. 1. Design, implement, test, and debug a Java® applet. 2. Use graphics in a Java® program.		
Reading	Read the Week Four Read Me First.		
Reading	Read Ch.13, "Applets and More," of Starting Out With Java.		
Participation	Participate in class discussion.	In class Tuesday, December 17, 2013	2
Supporting Activity Individual Graded In-Class Activity	This will be an in-class activity reviewing the week's discussion.	Assignment Tab or Submit to Facilitator Tuesday, December 17, 2013 10:00 PM	2

Week Four: Applets and Graphics			
	Details	Due	Points
Learning Team Instructions Initial Program	For the Initial Program at a minimum the GUI should have been coded. The program should compile and execute. Submit the .java source file for this program.	Assignment Tab Tuesday, December 17, 2013 6:00 PM	5

Week Four: Applets and Graphics			
	Details	Due	Points
Individual File Input Program	Write a program that is an application not an applet, written in Java with a graphical user interface. Using a text editor, create a file containing at least five lines of text. Write a Java application that reads the contents of the file and writes the lines from the file to a text area. The file should reside in the same directory as the .class file. The file should be referenced by the file name, not by an absolute path. Use an exception handler so that the program does not crash if the file is not found. At a minimum the application should contain a JTextArea. Submit the .java source code files with the correct file names for all classes used. The program must consist of at least one public class. Programs are expected to compile and run/execute correctly. There should be proper documentation (comments) in the source code. The documentation should include a block containing the name of the program and the name of the student, and a short description of the program.	Assignment Tab Tuesday, December 17, 2013 6:00 PM	15

Questions?

Week Four: Applets and Graphs

JComboBox

JComboBox

- Part of the java.swing package
- The JComboBox class is used to create an object where users select a choice from a list of choices
- JComboBox constructors and methods include:

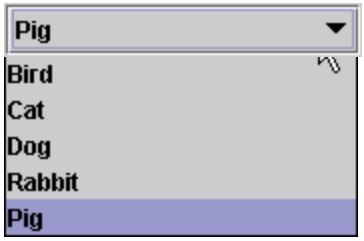
```
JComboBox()
setSelectedIndex()
getSelectedItem()
setEditable()
addActionListener()
```

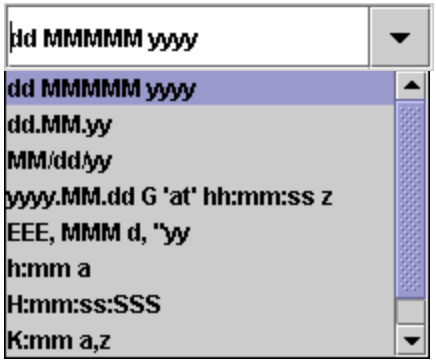
ComboBox

A JComboBox , which lets the user choose one of several choices, can have two very different forms.

uneditable combo box: features a button and a drop-down list of values. *(default form)*

editable combo box: features a text field with a small button abutting it. The user can type a value in the text field or click the button to display a drop-down list.





ComboBox Controls

The following code creates an uneditable combo box and sets it up:

```
String[] petStrings = { "Bird", "Cat", "Dog",
                        "Rabbit", "Pig" };

//Create the combo box, select item at index 4.
//Indices start at 0, so 4 specifies the pig.

JComboBox petList = new JComboBox(petStrings);
petList.setSelectedIndex(4);
```

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;      // we need event classes

// The DemoPizzaList class demonstrates writing to a textarea

public class DemoPizzaList extends JFrame implements ActionListener{

 //Make a scrolling text area
 JTextArea myTextArea = new JTextArea("Customer Orders", 3,30);
 //add textarea to the scrolling pane
 JScrollPane myScrollPane = new JScrollPane(myTextArea);

 JButton myOrderButton = new JButton("Make Order");   // create button
 JPanel northPanel = new JPanel();
 JPanel centerPanel = new JPanel();

 JTextField nameField = new JTextField (15);

 //Make a combobox
 String[] pizzaList = {"Make Selection", "Small $5", "Medium $10", "Large $20" };
 JComboBox myComboBox = new JComboBox<String>(pizzaList);
```

```
DemoPizzaList (){                              // the constructor
   setTitle("My Pizza Orders");
   setLayout(new BorderLayout());  //Layout for the Frame

   //configure north panel
   northPanel.setLayout(new GridLayout(3,3));
   northPanel.add(new JLabel ("Enter Name"));
   northPanel.add(nameField);
   northPanel.add(myComboBox);
   northPanel.add(myOrderButton);// add button
   myOrderButton.addActionListener(this);          // add listener
   add(northPanel,BorderLayout.NORTH);//add northPanel to frame

   //configure center panel
   myScrollPane.setVerticalScrollBarPolicy(
                  JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
   myTextArea.setEditable(false);
   centerPanel.add(myScrollPane);    // add scrollpane
   add (centerPanel,BorderLayout.CENTER); //add centerPanel to frame
   setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  }
```

```
public void actionPerformed(ActionEvent e) {     // the actionPerformed method
    String customerName = nameField.getText();
    if (customerName.length()==0 && myComboBox.getSelectedIndex() ==0){
        JOptionPane.showMessageDialog(null, "Please enter your order.");
    }
       myComboBox.setSelectedIndex(0);
 }

 public static void main(String[] args) {
   DemoPizzaList frame = new DemoPizzaList();
   frame.setSize(600,300);
   frame.pack();
   frame.setLocation(300,330);
   frame.setVisible(true);

 }
} // end of the DemoPizzaList class.
```

Group Exercise 1

- Modify the Pizza program so that it requires both the name of the user and the selection
- Add the Order informatio to the text area
- Only the actionPerformed method has to be modified

Questions?

Week Four: Applets and Graphics

Applets and Graphics

What Are Applets?

- An *applet* is a special Java program that can be embedded in HTML documents.
- It is automatically executed by (applet-enabled) web browsers.
- In Java, non-applet programs are called *applications*.

Application vs. Applet

- Application
  - Trusted (i.e., has full access to system resources)
  - Invoked by Java Virtual Machine (JVM, `java`), e.g.,
    `java HelloWorld`
  - Should contain a main method, i.e.,
    `public static void main(String[])`
- Applet
  - Not trusted (i.e., has limited access to system resource to prevent security breaches)
  - Invoked automatically by the web browser
  - Should be a subclass of class `java.applet.Applet`

Examples

- `HelloWorld.java`

```
public class HelloWord {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

- `HelloWorldApplet.java`

Compiling and Running

- To compile

  `javac HelloWorldApplet.java`

  Produces

  `HelloWorldApplet.class`

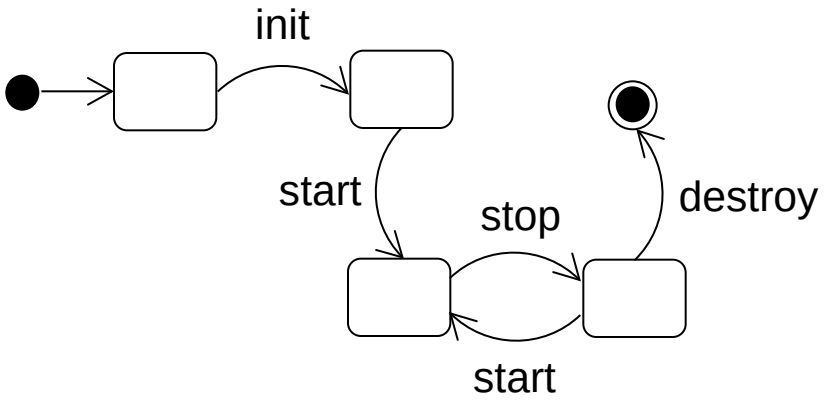
- To run
  - Open page `HelloWorld.html`from web browser or
  - Use appletviewer of JDK

    `appletviewer HelloWorld.html`

Elements of Applets

- No `main` method
- `paint` method to paint the picture
- Applet tag: `<applet>` `</applet>`
  - `code`
  - `width` and `height`

The Life-Cycle of Applet



The Life-Cycle of Applet

- init()
  - Called exactly once in an applet's life.
  - Called when applet is first loaded, which is after object creation, e.g., when the browser visits the web page for the first time.
  - Used to read applet parameters, start downloading any other images or media files, etc.

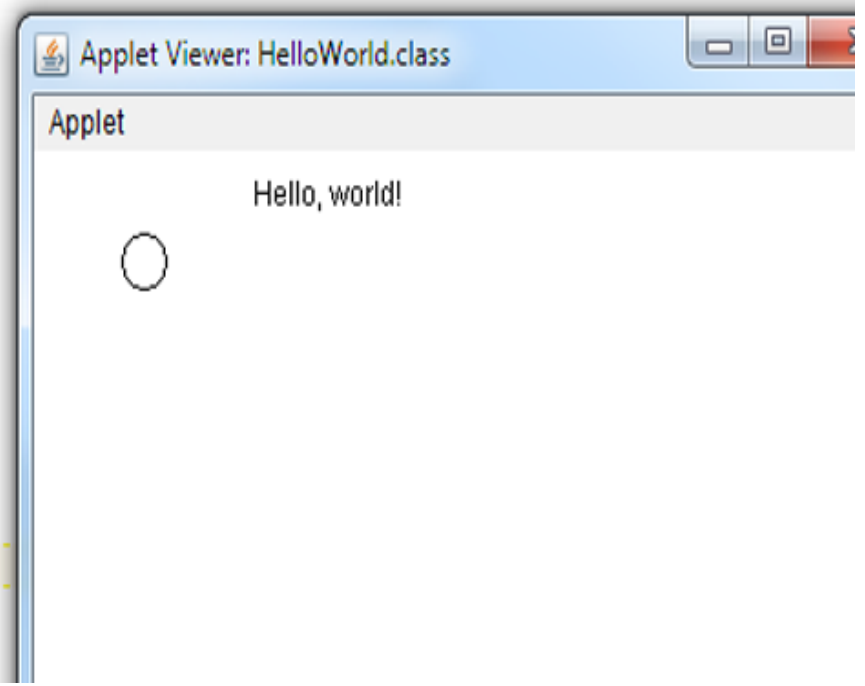
Applet Life-Cycle (Cont.)

- start()
  - Called at least once.
  - Called when an applet is started or restarted, i.e., whenever the browser visits the web page.
- stop()
  - Called at least once.
  - Called when the browser leaves the web page.

Applet Life-Cycle (Cont.)

- destroy()
  - Called exactly once.
  - Called when the browser unloads the applet.
  - Used to perform any final clean-up.

```java
import java.awt.*;
import javax.swing.*;

// Applet code for the "Hello, world!" example.
// This should be saved in a file named as "HelloWorld.java".
public class HelloWorld extends JApplet {
    public void init() {

        JOptionPane.showMessageDialog(null, "Initializing");
  }
    public void stop() {

        JOptionPane.showMessageDialog(null, "Stopping");
   }
   // Print a message on the screen (x=20, y=10).
    public void paint(Graphics g) {
    JOptionPane.showMessageDialog(null, "Painting");

        g.drawString("Hello, world!", 100, 20);

        // Draws a circle on the screen (x=40, y=30).
        g.drawArc(40, 30, 20, 20, 0, 360);
    }
}
```



Methods for Drawing

- Rectangles: drawRect(…); fillRect(…);
- Ovals: drawOval(…); fillOval(…);
- Arcs: drawArc(…) fillArc(…);
- Line: drawLine(…);

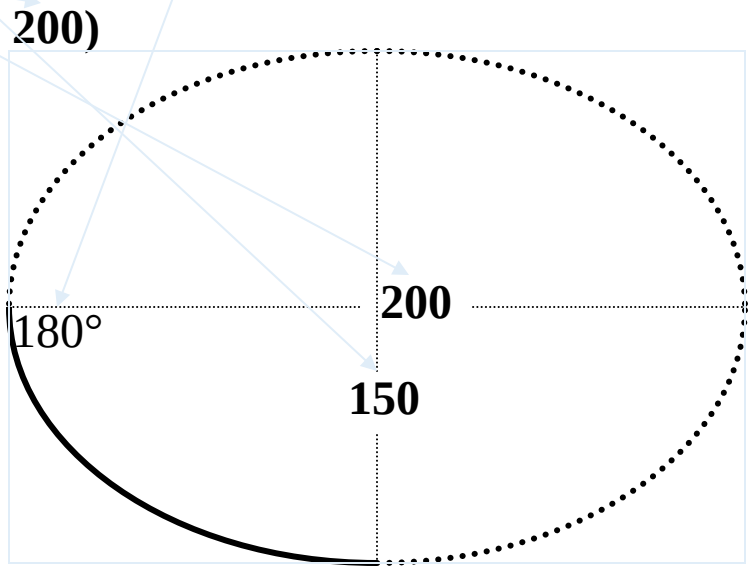
- Details about the input parameters for of these methods can be found in the

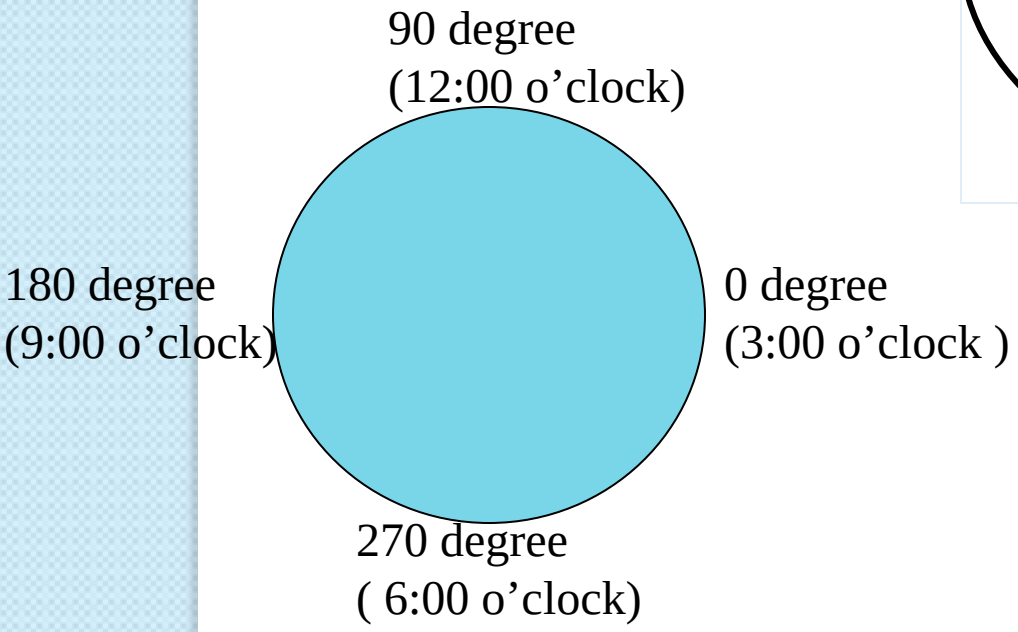
For example:
drawArc (6 parameters)

- X position
- Y position
- Width
- Height
- Starting point (360 degree of Clock)
- Total Angle of Arc

g.drawArc(100, 200, 200, 150, 180, 90)

x start, y start, x diameter, y diameter, start angle, arc angle





COLORS STANDARD 13

- 256 * 256 * 256 (Lots of possible)
  - black
  - blue
  - cyan
  - darkGray
  - gray
  - green
  - lightGray
  - magenta
  - orange
  - pink
  - red
  - white
  - yellow

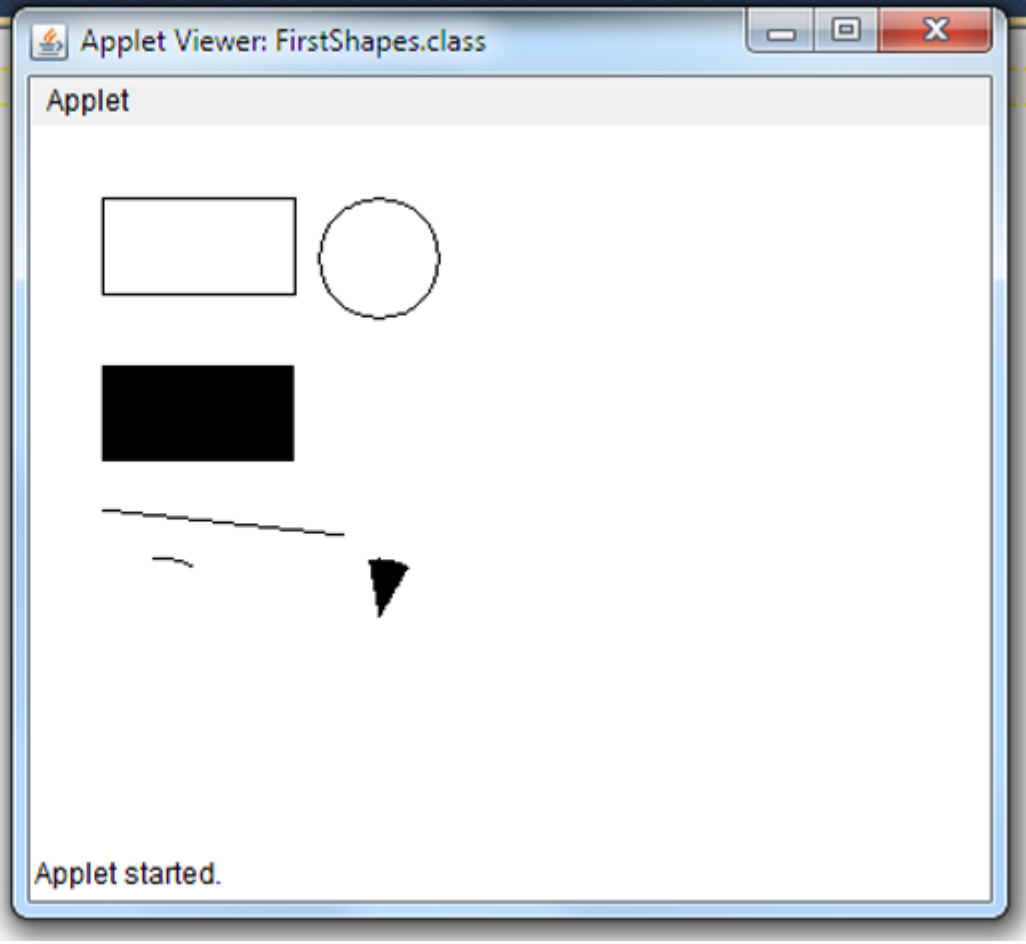
g.setColor(Color.red);

Sequence of "Graphics" Applet

```
import java.awt.*;
import java applet.Applet;

public class FirstShapes extends Applet {
   public void paint (Graphics g) {
     g.drawRect (30,30,80,40);
     g.drawOval(120,30,50,50);
     g.setColor(Color.black);
     g.fillRect(30,100,80,40);
     g.drawLine (30,160,130,170);
     g.drawArc (30,180,50,50,60,40);
     g.fillArc(120,180,50,50,60,40);
   }
}
```

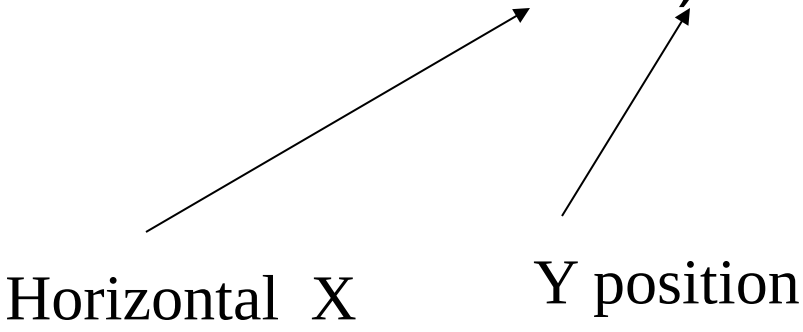
FirstShapes.java | PieChart.java | Document1

```
import java.awt.*;
import java.applet.Applet;

public class FirstShapes extends Applet {
public void paint (Graphics g) {
g.drawRect (30,30,80,40);
g.drawOval(120,30,50,50);
g.setColor(Color.black);
g.fillRect(30,100,80,40);
g.drawLine (30,160,130,170);
g.drawArc (30,180,50,50,60,40);
g.fillArc(120,180,50,50,60,40);
}
}
```



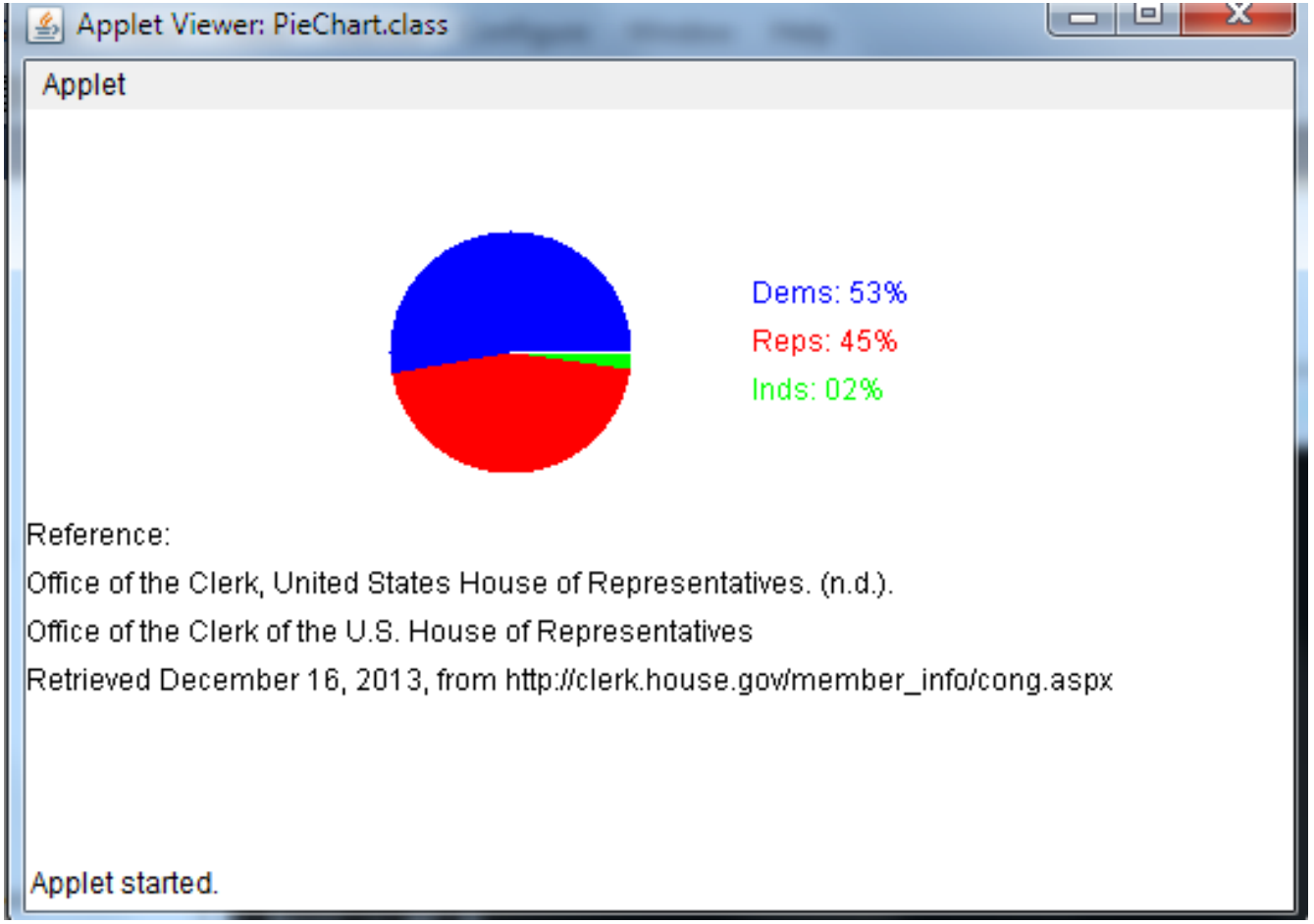
Add some text…

- g.drawString (“ Hello World “, 30, 40);



Concatenation + operator

- g.drawString(“Hi” + “there” + “Mike” ,100,100);


Applet Viewer: PieChart.class
Applet
Dems: 53%
Reps: 45%
Inds: 02%
Reference:
Office of the Clerk, United States House of Representatives. (n.d.).
Office of the Clerk of the U.S. House of Representatives
Retrieved December 16, 2013, from http://clerk.house.gov/member_info/cong.aspx
Applet started.

```
/*
 Draws a pie chart, given the numbers of Democrats, Republicans, and
 Independents in the US Senate:
 53 Democrats
 45 Republicans
 2 Independents

 Office of the Clerk, United States House of Representatives. (n.d.).
 Office of the Clerk of the U.S. House of Representatives
 Retrieved December 16, 2013, from http://clerk.house.gov/member_info/cong.aspx
 */
   import java.awt.*;
   import javax.swing.JApplet;
   import java.text.DecimalFormat; // Formats number

public class PieChart extends JApplet
{
   DecimalFormat df = new DecimalFormat("00");

   public void paint(Graphics g) {

   int Dems, Reps, Inds, Total; // # of Senators in each party and the total
```

```
float PercReps, PercDems, PercInds;  // The percentages

int x = 150, y = 50, w = 100, h = 100;   // defines the size of the pie
int startAngle, degrees; // will be used to draw a pie slice

// Set the # of Senators in each party
Dems = 53;
Reps = 45;
Inds = 2;

// Compute percentages
Total = Dems + Reps + Inds;

// %Dems
PercDems = (Dems * 100.0f) / Total;

// %Reps
PercReps = (Reps * 100.0f) / Total;

// %Inds
PercInds = (Inds * 100.0f) / Total;
```

```
// Display/Output results
System.out.println("% Dems = " + PercDems);
System.out.println("% Reps = " + PercReps);
System.out.println("% Inds = " + PercInds);

// Display the Pie Chart

// Draw the Arc for Dems
startAngle = 0;
degrees = (int)(PercDems * 360 / 100);
g.setColor(Color.blue);
g.fillArc(x, y, w, h, startAngle, degrees);
//Write legend for Dems
g.drawString("Dems: " + df.format(PercDems) + "%", 300, 80);

// Draw the Arc for Reps
startAngle = degrees;
degrees = (int)(PercReps * 360 / 100);
g.setColor(Color.red);
g.fillArc(x, y, w, h, startAngle, degrees);
//Write legend for Reps
g.drawString("Reps: " + df.format(PercReps) + "%", 300, 100);
```

```
    // Draw the Pie for Inds
    startAngle = startAngle + degrees;
    degrees = (int)(PercInds * 360 / 100);
    g.setColor(Color.green);
    g.fillArc(x, y, w, h, startAngle, degrees);
    //Write legend for Inds
    g.drawString("Inds: " + df.format(PercInds) + "%", 300, 120);

    g.setColor(Color.black);
    g.drawString ("Reference:",0, 180);
    g.drawString ("Office of the Clerk, United States House of Representatives. (n.d.).", 0, 200);
    g.drawString ("Office of the Clerk of the U.S. House of Representatives", 0, 220);
    g.drawString ("Retrieved December 16, 2013, from http://clerk.house.gov/member_info/cong.aspx", 0, 240);

    } // paint

} // PieChart
```

Group Exercise - 2

Create a Pie Chart Applet to show the following

In October 2013, the usage of browsers among users of the W3Schools Website was:

- Internet Explorer 11.7%
- Firefox 27.2%
- Chrome 54.1 %
- Safari 3.8%
- Opera 1.7%

- Browser Statistics. (n.d.). *Browser Statistics*. Retrieved from http://www.w3schools.com/browsers/browsers_stats.asp

Questions?

Week Four: Applets and Graphics

Exception -- NumericException

```
// An exception-handling example
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class NumericInput extends JFrame
   implements ActionListener {

   private JTextField inputField;
   private JButton readButton;

   // set up GUI
   public NumericInput()
   {
      setTitle( "Demonstrating Exceptions" );

      setLayout( new GridLayout( 2, 2 ) );

      // set up label and inputField
     add(new JLabel( "Enter number "));
     inputField = new JTextField (10);
     add(inputField);
```

```
        //Add button
        readButton = new JButton ("Enter Number");
        add (readButton);
        readButton.addActionListener(this);

        setSize( 425, 100 );
        setLocationRelativeTo(null);
        setVisible( true );

    } // end constructor

    // process GUI events
    public void actionPerformed( ActionEvent event )
    {

        // read input
        try {
            double number1 = Double.parseDouble( inputField.getText() );
            JOptionPane.showMessageDialog( null, "Thank You, GoodBye");
            System.exit(0);  //Close Program
        }
```

```
      // process improperly formatted input
      catch ( NumberFormatException ne ) {
         JOptionPane.showMessageDialog( null, "Enter a valid number");
      }

   } // end method actionPerformed

   public static void main( String args[] )
   {
      NumericInput application = new NumericInput();
      application.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
   }

} // end class NumericInput
```

Group Exercise 3

- Modify the revised Pizza program by adding a numeric field to record the number of pizzas ordered
- Use the NumberFormatException to ensure that this is a numeric entry
- Add the number order information to the other order information in the textarea. For example:
- Bob Small $5 2

Questions?

Class Topics

- Week One: User Interfaces
- Week Two: GUI Components
- Week Three: Files
- Week Four: Applets and Graphics
- Preview
- Week Five: Database Connectivity and Mobile Computing

Week Five: Database Connectivity and Mobile Computing			
	Details	Due	Points
Objectives	1. 1. Design, implement, test, and debug a Java® program that connects to a database. 2. Discuss considerations in writing programs for mobile devices with the Java® language.		
Reading	Read the Week Five Read Me First.		
Reading	Read Ch. 15, "Databases," of Starting Out With Java.		
Reading	Read Ch. 28, "Accessing Databases with JBDC," of Java: How to Program.		
Reading	Read Hour 24, "Writing Android Apps," of Sams Teach Yourself Java in 24 Hours.		
Reading	Read Appendix D, "Setting up an Android Environment," of Sams Teach Yourself Java in 24 Hours.		
Participation	Participate in class discussion.	In class Tuesday, January 07, 2014	2
Supporting Activity Individual Graded In-Class Activity	This will be an in-class activity reviewing the week's discussion.	Assignment Tab or Submit to Facilitator Tuesday, January 07, 2014 10:00 PM	2

Week Five: Database Connectivity and Mobile Computing			
	Details	Due	Points
Objectives	1. Design, implement, test, and debug a Java® program that connects to a database. 2. Discuss considerations in writing programs for mobile devices with the Java® language.		
Reading	Read the Week Five Read Me First.		
Reading	Read Ch. 15, "Databases," of Starting Out With Java.		
Reading	Read Ch. 28, "Accessing Databases with JBDC," of Java: How to Program.		
Reading	Read Hour 24, "Writing Android Apps," of Sams Teach Yourself Java in 24 Hours.		
Reading	Read Appendix D, "Setting up an Android Environment," of Sams Teach Yourself Java in 24 Hours.		
Participation	Participate in class discussion.	In class Tuesday, January 07, 2014	2
Supporting Activity Individual Graded In-Class Activity	This will be an in-class activity reviewing the week's discussion.	Assignment Tab or Submit to Facilitator Tuesday, January 07, 2014 10:00 PM	2

Week Five: Database Connectivity and Mobile Computing			
	Details	Due	Points
Learning Team Final Program	Create a GUI-based program to accept name of donor, name of charity, and amount of pledge from the user. The name of the donor and the amount of the pledge should be entered in textfields. The name of the charity should be selected from a menu. The program should contain error checking. At a minimum, the name field should not be blank and the amount field should be numeric. The amount field should be in numeric format. After the information is entered, the user should press a JButton causing the entry to be displayed in a JTextArea. Submit the .java source code files with the correct file names for all classes used. The program must consist of at least one public class. Programs are expected to compile and run/execute correctly. There should be proper documentation (comments) in the source code. The documentation should include a block containing the name of the program and the name of the student, and a short description of the program. Be prepared to demonstrate your program to the class.	Assignment Tab Tuesday, January 07, 2014 6:00 PM	10

Week Five: Database Connectivity and Mobile Computing			
	Details	Due	Points
Individual Pie Chart Applet	Write an applet that includes a pie chart. Use a news article with statistics that are good candidates for a pie chart: for example, political candidate preferences; percentages of those for, against, or undecided about a ballot measure; and so forth. Cite the source for your input statistics using APA format. Submit the applet along with an HTML file to launch it. Submit all .java files, specifically the source code for the applet. Submit the .java source code files with the correct file names for all classes used. The program must consist of at least one public class. Programs are expected to compile and run/execute correctly. There should be proper documentation (comments) in the source code. The documentation should include a block containing the name of the program and the name of the student, and a short description of the program.	Assignment Tab Tuesday, January 07, 2014 6:00 PM	15

Questions?

Questions?

See You January 7!!