DecimalFormat & NumberFormat Class Notes

 The following presents two techniques for formatting numbers to decimal. Technique 1 uses the DecimalFormat class. Technique 2 uses the NumberFormat class. Both techniques require proper imports as listed in the examples below.

The DecimalFormat class can be used for simple formatting. Please do not be afraid to Google or search the Java API to learn all the methods and capabilities of DecimalFormat. However here is a simple way to format floats, doubles and longs into a typical format.

You must create an instance of the DecimalFormat class and create a format of the form (#,###.##). This is just an example that will display a number in the thousands with a comma and 2 decimal places (i.e. 3,450.21). You can use any format to your liking. Once your DecimalFormat Object has been instantiated with your chosen format you can simply use that object with the format method and pass in a double value or a long value which will return a String of your number with the expected format.

I have listed an example blip of code below:

import java.awt.\*;

import java.applet.Applet;

import java.awt.event.\*;

**import java.text.DecimalFormat;**

public class FormattingNumbers extends Applet implements ActionListener

{

 //Declarations

 TextField tfNumber1, tfNumber2;

 int iNumber1, iNumber2;

 double dQuotient;

 **String sOutput;**

 //Declare a format for outputing numbers with commas and decimal points

 **DecimalFormat MyFormat = new DecimalFormat("#,###.##");**

 public void init()

 {

 //create and add a text field for number1 to the applet

 tfNumber1 = new TextField("Enter Number 1 Here", 10);

 add(tfNumber1);

 //create and add a text field for number2 to the applet

 tfNumber2 = new TextField("Enter Number 2 Here", 10);

 tfNumber2.addActionListener(this);

 add(tfNumber2);

 }

 public void paint( Graphics g )

 {

 g.drawString("The Quotient of Number1/Number2 is " + **sOutput**,10,100); //output

 }

 public void actionPerformed( ActionEvent e )

 {

 //get number from the users input

 iNumber1 = Integer.parseInt(tfNumber1.getText());

 iNumber2 = Integer.parseInt(tfNumber2.getText());

 //compute the quotient of the two numbers

 dQuotient = (double)iNumber1/iNumber2;

 //convert your quotient into an output string of the format you declared

 **sOutput = MyFormat.format(dQuotient);**

 repaint();

 }

}

A second technique that can be used to display a double in US currency form requires the usage of the **NumberFormat.getCurencyInstance( )** method. Your code would looks as follows:

Include this import:

import java.text.NumberFormat;

Assume money contains the number to format:

double money = 1323.526;

Declare globally and instantiate the following object:

NumberFormat currency = NumberFormat.getCurrencyInstance( );

Then in Paint, to format and display your value:

g.drawString(“US:” + currency.format(money), 70, 90);

Amount displayed = $1,323.53