Presenting Data to Users:

- When we print data out from our programs, we need to think about the users. This is why we want to use appropriate labels and space everything out well. Another reason for making our data more comprehensible to users is formatting our numbers nicely. Java provides a couple different ways to do this.

DecimalFormat Class:

- This allows us to specify our format for floating point numbers.

- We can specify the # of digits before or after the decimal point.

- Specify commas

- Specify dollar signs

- Specify percentage, which will print a percent sign but also multiply the value by 100.

- and others

Preparing to use DecimalFormat Class:

- In the package java.text

import java.text.DecimalFormat;

- After this, we need to create a String that holds the pattern we want.

- Declare and create a DecimalFormat object that uses the pattern String we want to use. In the parentheses we will pass the String that is the pattern we want to use.

DecimalFormat myFormat = new DecimalFormat(pattern);

- Now that we have the formatting object, we’ll want to actually use it. What we use is the .format method of the DecimalFormat class. We pass in the parameter the number that will be formatted. That’s going to apply the pattern and give us a string that we can print or use in some way.

String format(double number)

String numStr = myFormat.format(1103.2);

System.out.println(numStr);

After creating a DecimalFormat object, we will use the method

1. format, which takes a number as a parameter and prints the formatted number to System.out.
2. format, which takes a number as a parameter and return a String containing the formatted number.
3. setValue, which takes a number as a parameter, and set the object to format that number.

- We can specify patterns manually, but Java also provides us with a shortcut called NumberFormat class.

The NumberFormat class has static methods that return a NumberFormat object which will correctly format currency or percents.

static NumberFormat getCurrencyInstance();

static NumberFormat getPercentInstance();

String format(double number);

- When it comes to NumberFormat and getCurrencyInstance, these are very useful for localization purposes. Sometimes when we write our programs, the programs will be used in a lot of different places around the world. Of course, other places in the world use different currencies in their standard operations. We can specify the locale to format for a different currency by checking the documentation. There are different versions of getCurrencyInstance which will format for different types of currency.