

Using library methods – posted 9/21, due 10/3

This is just a short lab, as an opportunity for you to learn some helpful and useful new library method calls. We will make extensive use of the ability to "copy" classes into objects; that is, when we find a useful class, we can create a local copy of it using the "new" statement, in order to use its methods.

1. Start with a program framework, and call it Lab4MainClass. Import THREE LIBRARIES:

```
import javax.swing.JOptionPane;
import java.util.Random;
import java.util.Date;

public class Lab4MainClass
{
public static void main( String args [ ] )
{

    } // end main
} // end class
```

2. In main (as usual) go to the user, and get a String. You will have to create a String variable, and make a call to JOptionPane. However, first CREATE AN OBJECT of the JOptionPane class. Here's the code:

```
JOptionPane myMenus = new JOptionPane();
String userInput = " ";
userInput = myMenus.showInputDialog("Enter a number between 1 and 100: ");
```

Answer these questions in comments in your code:

Question 1: (place the answer in comments in your code): What is the Object Name?

Question 2: What is the Class name, from which the object is derived?

Question 3: What method is being called?

Question 4: What parameter is being sent to the method?

Question 5: What is returned from the method?

3. Now, userInput will contain a String, which is useless in a math equation. For instance, try using userInput in a math equation, and see the result.

```
int x = 10;
int sum = 0;

sum = userInput + x;
System.out.println( sum );
```

Question 6: what happens?

4. To fix the inconsistency, we need to convert userInput to an integer. It's rather easy, by using the parseInt() method of the Integer class:

```
int y = 0;  
y = Integer.parseInt( userInput );
```

in this example, Integer is a class, parseInt is a method, userInput is the parameter sent to the method (a String), and y is the integer that results. y can now be used in a math equation, since it is a true integer.

Your program should run fine if you enter any number.

Question 7: What happens if you enter the word "cat" instead of a number when the program runs? (remember, answer these questions in comments in your code)

6. Get a Random number for x. To do this, we will make use of the nextInt() method of the Random class. Create an OBJECT of the Random class called myGen, and use the object to access the nextInt() method:

```
Random myGen = new Random( );  
x = myGen.nextInt( );  
System.out.println("Random number x is: " + x );
```

now use x in any math equation.

7. Let's use another class to print today's date. It's easy, just create an object of the Date class, and print it! That's because someone took the time to create a really powerful class that does all the work for us.

```
Date todaysDate = new Date( );  
System.out.println( todaysDate );
```

8. Finally, let's use the library class Math to perform some math equations. Try these methods:

Math.pow (double a, double b) - Returns the value of the first argument raised to the power of the second argument.

Math.sqrt(double a) - Returns the positive square root of a double value.

Math.max(a, b)

Math.min(a, b)

Question 8: what do they do? (remember, answer these questions in comments in your code)

Final Note: you've probably noticed some class that are copied (JOptionPane, Random, Date) and some that aren't (System, Math). Whether a class can be copied or not, is at the discretion of the author. The copy operation is called INSTANTIATION.

Here is a web site that summarizes all of Java classes and methods: <http://leepoint.net/notes-java/>
I have no idea where this is from.