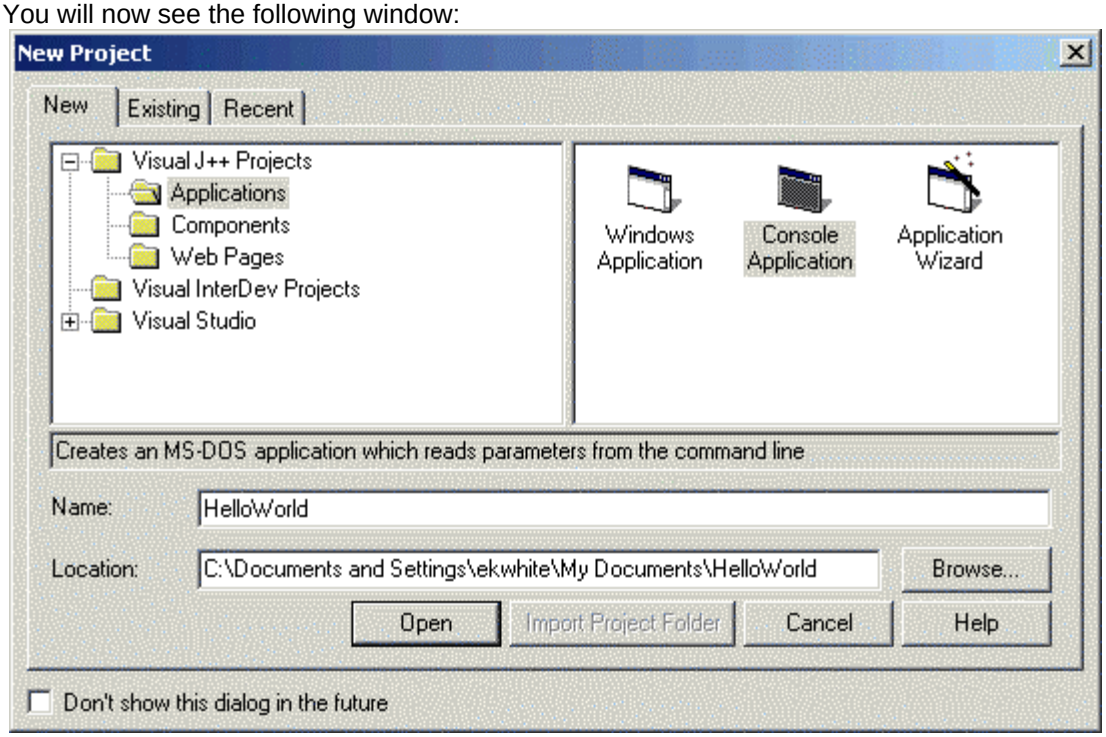
Recitation 1, 01/27/2003
Instructions for creating your first program

This exercise will walk you through setting up a simple input-output program in Java. Your homework assignment will use similar logic to set up an area calculation. In general, we make the recitation program similar to (but easier than) the homework assignment for that week.

Step 1 (Start up Visual J++):

- Select the **Start** button (bottom left side of the screen).
- Select **Programs.**
- Select **Microsoft Visual J++ 6.0.**
- And again select **Microsoft Visual J++ 6.0.** (Redundancy is often built into Windows applications.)

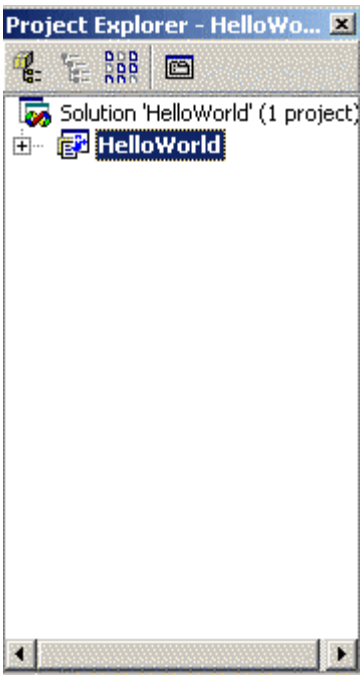
You will now see the following window:



Step 2 (Start a new project):

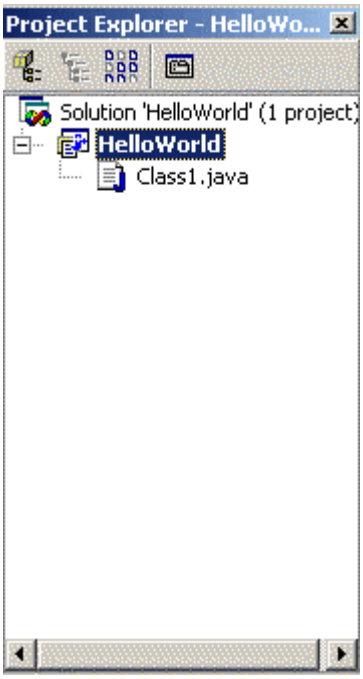
- Make sure the **New** tab is selected. This starts a completely new project.
- In the left pane select **Application.** This creates a standalone program.
- In the right pane select **Console Application**. This means that when your program runs, you will see its output in a black DOS-like window with gray text.
- In the **Name** field (in the lower part of the window), replace ***Project1*** with ***HelloWorld.***
- In the **Location** field, enter the location where your program will live for now, for example, ***C:\Documents and Settings\<your login name>\My Documents\HelloWorld***). Your login name is whatever name you use to log onto the machines in this lab. Do not remove the ***HelloWorld*** at the end of the path name, because that is the name of the file.
- Now click the **Open** button.

You will now see the following window, marked ***Project Explorer***, in the corner of your screen:

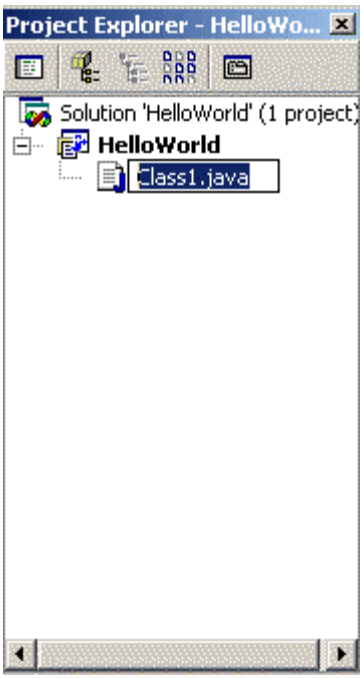


Step 3 (Change the file name):

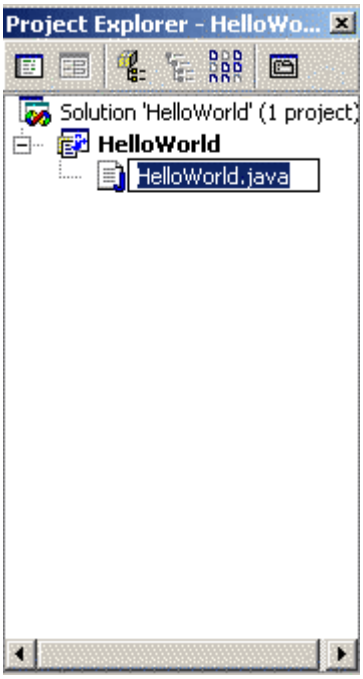
- Double click HelloWorld (the blue highlighted name in the above window), to find out what java code file has been created. The window now looks like this:



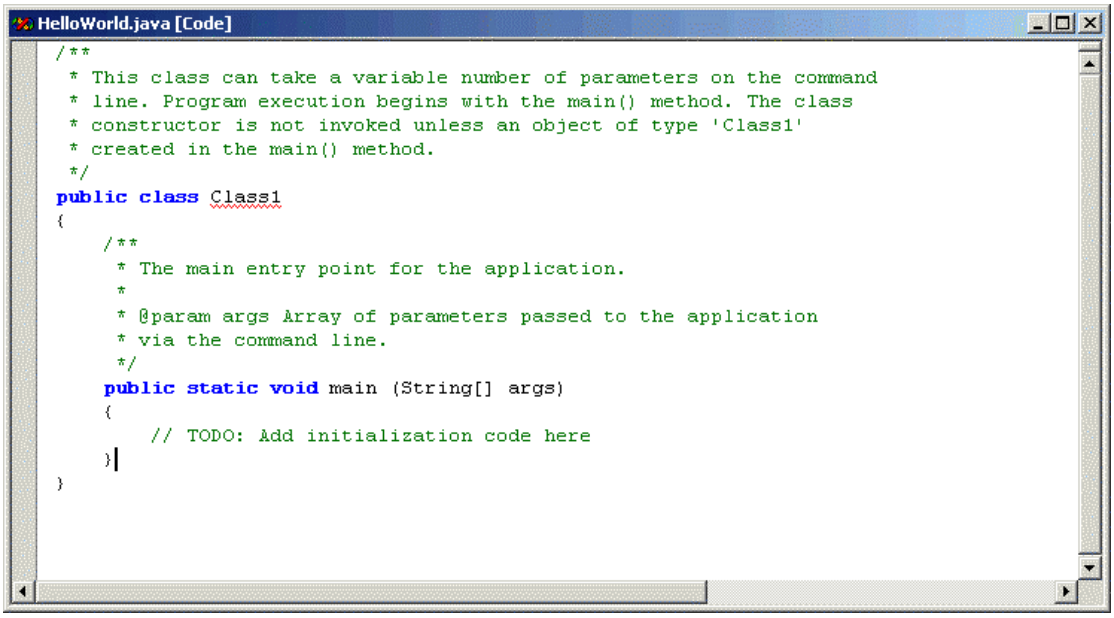
- In Project Explorer, highlight and right-click Class1.java to rename it.
- On the shortcut menu, click **Rename.** The window looks like this



- Type the new name HelloWorld.java, and press Enter. The window should look like the one below. Don't forget the .java extension! This lets Microsoft J++ know that you are writing a piece of java code.



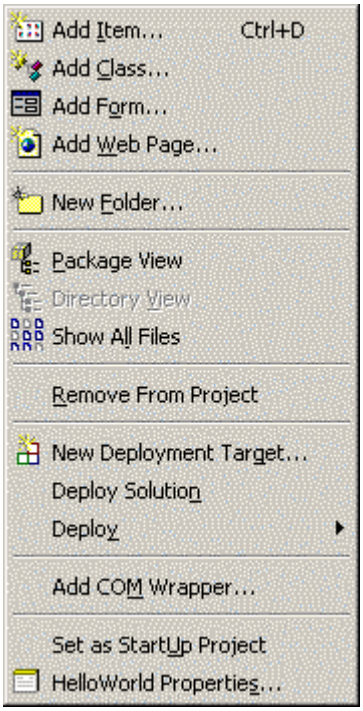
- In Project Explorer, double-click HelloWorld.java. You'll see a new window, with the java code started for you.



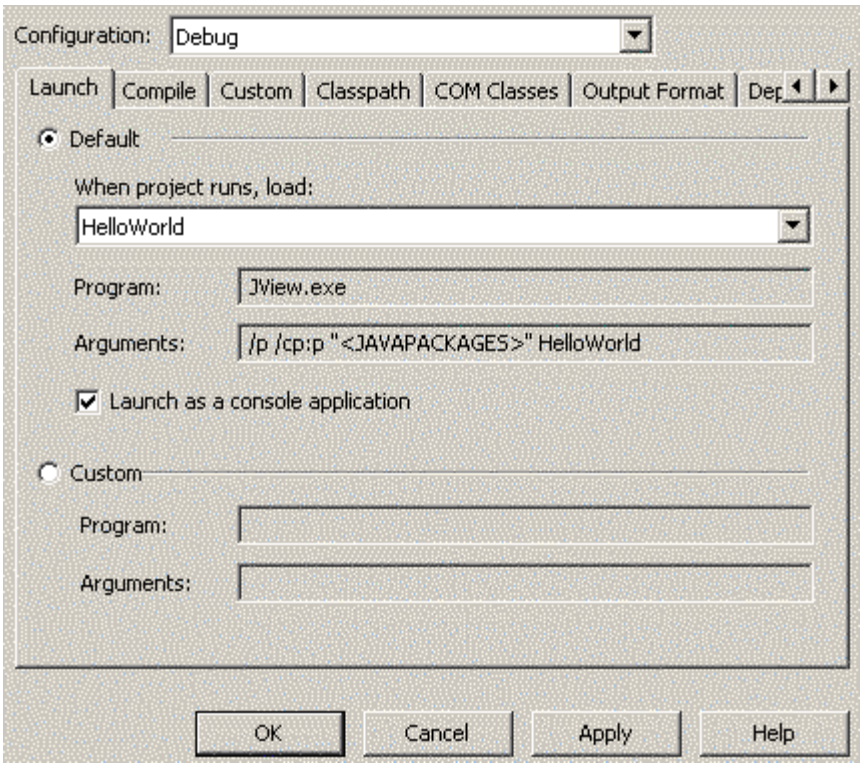
· Change the class name, which is Class1, to HelloWorld. It's important for this class to have the same name as the .java file; that's why you have to change them both.

Step 4 (set up the properties of your project)

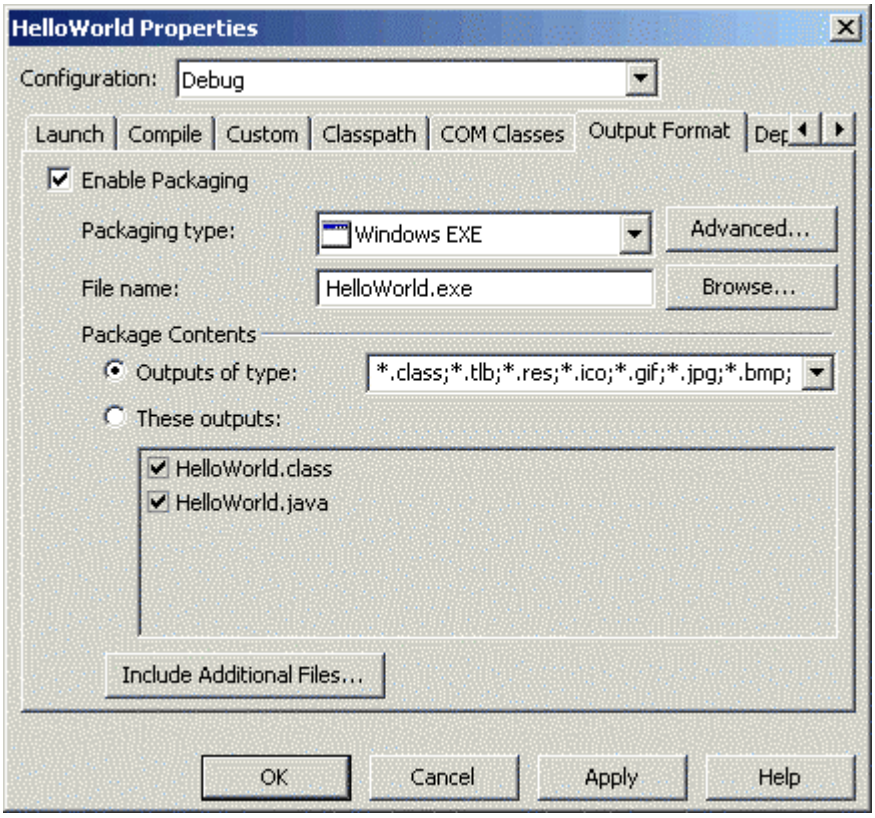
· From the menu bar, select the **Project** drop-down menu. Click on **HelloWorld Properties**, the choice at the bottom of the menu, as in the menu below.



A new window will appear.



· Choose the Output Format tab. You should see two filenames: HelloWorld.class and HelloWorld.java.



· Choose Apply and OK.
· Remember this step for later projects! When you start a new project, you will often see an error message “Specified main Class1 not found” when you try to compile the code into a running program. Making sure the project properties are set correctly and that the

.class and .java files are present, as in the step above, will nearly always solve this problem.

Step 5 (add some Java code to your project)

- At the top of your program add comments, such as the following lines (substitute your name).

```
/* Your name
A Hello World program */
```

Notice that any text you type between /* and */ is colored green; this is Java's way of telling you that it knows that these lines are comments, not actual code.

- Add the following line to the 'main' method:

```
import java.io.*;
```

in the very first line of the program after the green comments. This line makes the Java input/output functions available for use in the program. The semicolon is how you end a statement in Java, just like a period ends a sentence in English.

- The line

```
public static void main (String [] args)
```

defines the place where the program begins running.

- Modify the 'main' method with two extra words, so that it reads:

```
public static void main (String[] args) throws IOException
```

Exceptions are how Java handles weird code behavior or unexpected errors at run time.

- Add the following line to the 'main' method:

```
BufferedReader stdin = new BufferedReader
    (new InputStreamReader(System.in));
```

Don't worry about exactly what this code is doing yet. It just prepares the program to accept input from the keyboard—take it on faith, for now.

- After this, type in:

```
int birthyear, age;
```

This tells Java that you're planning to use two integer variables, one named birthyear and one named age, in the program.

Below that, type in

```
String name;
```

This lets Java know that you're planning to use a variable called name to store text in the program.

- Add the following line:

```
System.out.println("Hello World");
```

This Java statement prints out the text string "Hello World" to the screen. If you decide to copy and paste this text in, and Java complains, you may need to retype the quotes in the Java editor.

- Add the following lines right after the line that prints "Hello World" to the screen.

```
System.out.print("Enter your name: ");
name = stdin.readLine();
```

The readLine() method reads input from the keyboard, until the Enter key is pressed. Here, the code stores the text from the keyboard in the name variable.

Now add:

```
System.out.print("Enter your year of birth: ");
birthyear = Integer.parseInt(stdin.readLine());
```

The readLine() method reads input from the keyboard, until the Enter key is pressed. Here, the code converts text from the keyboard into an integer number and stores the number in the birthyear variable.

- Calculate the age from the year of birth:

```
age = 2003 - birthyear;
```

This line stores the difference between the year of birth and this year (2003) in the variable age.

- Add the following two output lines:

```
System.out.println("Hello " + name + "!");
System.out.println("This year, you will be " + age + " years old-—good luck!");
```

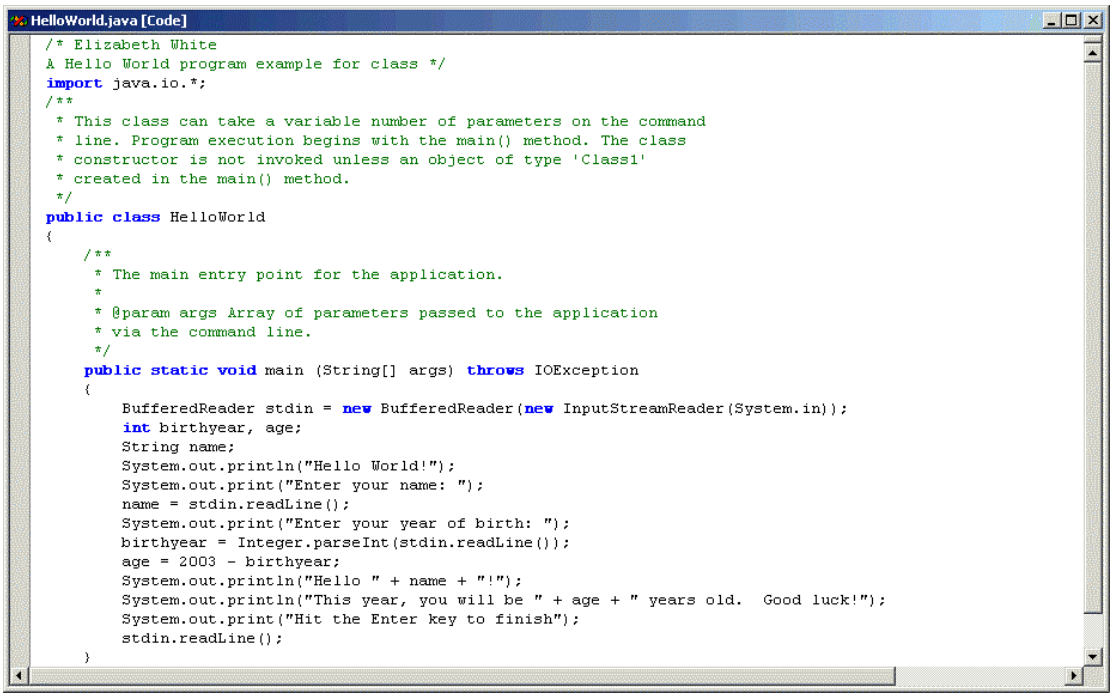
These two lines show how you how to print text strings and numbers back to the screen as your program runs. Notice that you have to include spaces.

- Add the following line to pause the program until the user hits the Enter key:

System.out.print(“Hit the Enter key to finish”);
stdin.readLine();

Once the program is actually running, the readLine() method pauses it until the user hits the Enter key. This lets you read the output of your program before its window closes.

- Your code should now look like this:



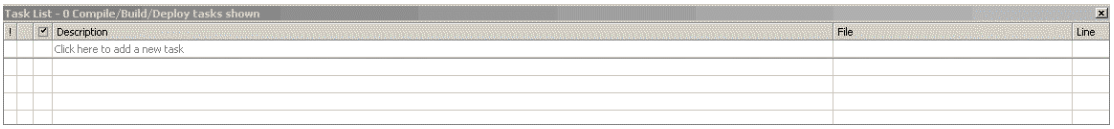
```
HelloWorld.java [Code]
/* Elizabeth White
A Hello World program example for class */
import java.io.*;
/**
 * This class can take a variable number of parameters on the command
 * line. Program execution begins with the main() method. The class
 * constructor is not invoked unless an object of type 'Class1'
 * created in the main() method.
 */
public class HelloWorld
{
    /**
     * The main entry point for the application.
     *
     * @param args Array of parameters passed to the application
     * via the command line.
     */
    public static void main (String[] args) throws IOException
    {
        BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));
        int birthyear, age;
        String name;
        System.out.println("Hello World!");
        System.out.print("Enter your name: ");
        name = stdin.readLine();
        System.out.print("Enter your year of birth: ");
        birthyear = Integer.parseInt(stdin.readLine());
        age = 2003 - birthyear;
        System.out.println("Hello " + name + "!");
        System.out.println("This year, you will be " + age + " years old.  Good luck!");
        System.out.print("Hit the Enter key to finish");
        stdin.readLine();
    }
```

Step 6 (Saving your work):

- From the menu bar, select **File,** then **Save All**. Save your work often!

Step 7 (Compile your program):

- From the menu bar, select **Build.** This converts your human-readable .java file to a format the computer can read easily. Look at the bottom of the screen, in this window, for any error messages.

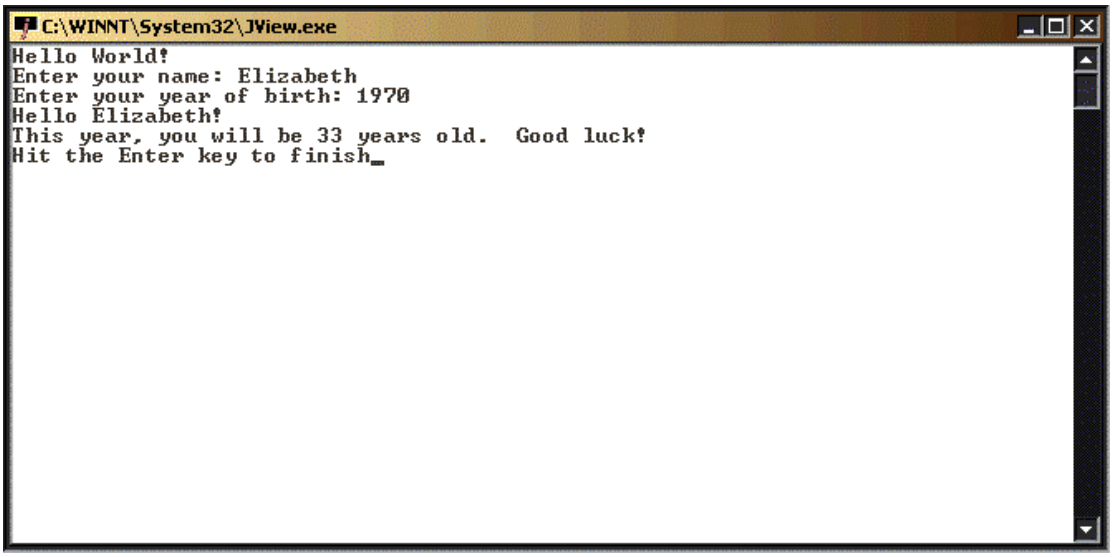


Task List - 0 Compile/Build/Deploy tasks shown

!		Description	File	Line
		Click here to add a new task.		

Step 8 (Run your program):

- From the menu bar, select **Debug** and then **Start** to start your program. Then a new black window will appear (it may be hidden behind another window on your screen, so look carefully). It should look like a color-inverted version of the window below. When you are satisfied that the code does what you expect, hit Enter to let the program stop running and close.



Step 10 (Save your program to disk):

- Minimize the Java program by clicking on the leftmost button, marked –, in the upper right hand corner. Go to My Computer, find your project folder (this is probably **C:\Documents and Settings\<*your login name*>\My Documents\HelloWorld**, or whatever you named it) and save all the files to your floppy disk or Zip disk. Then you should feel free to log off.