Java Primer

A step-by-step guide to creating a simple example program

This year, we are using a simple command line interface to compile our java programs. This will walk you through a simple example to show you how this works. At the end, you will have a working adding machine. ohhh... exciting...

1. Make a team directory and get the code.
   a. Go to C:\16x62\Teams and make a new directory with your team name.
   b. Copy the directory `C:\16x62\Samples\SampleBot` into your team directory.
   c. The files in the Samples directory are read-only. After you copy the SampleBot directory into your team directory, select all the files, right-click, select properties, and un-check 'read-only'. This will allow you to make changes.

2. Compile and run the sample code.
   a. Open a command window (Start->Run and then type 'command') and cd to your team's copy of SampleBot. TIP: If Windows Explorer is open to your directory, then command will start there automatically.
   b. At the command line, type:

      ```
      javac *.java
      ```

      This will compile all the java files (javac = java compiler). It should say:

      ```
      compiling: Applet1.java
      compiling: Robot.java
      compiling: RobotController.java
      compiling: SonarConsole.java
      ```

   c. Use the java run-time environment to actually run the program. Type:

      ```
      java Robot
      ```

      (we ran Robot because Robot.java has the main() in it) A 'Sample Bot!' window should appear. Right now, don't worry about what the buttons do.
   d. To quit the program, go to the command window and type `ctrl-c`

3. Modify the source code to make an adding machine.

   When you ran the SampleBot program above, you may have noticed a couple spare buttons named 'button3' and 'button4' along with two text fields 'textField1' and 'textField2'. We are now going to make these do something. We'll make button3 add the numbers in the two text fields and display the result. Believe it or not, this will not be the coolest programming assignment you do this year. Anyway, to the code...

   a. Open Robot.java in your favorite text editor (emacs, jext, wordpad, word, etc)
   b. All the code at the top sets up the window and buttons, handles mouse clicks, etc. You can just scroll down until you get to the `button*_MouseClick` functions that contain what happens when somebody actually clicks a button. Buttons 1 and

2 already have code, and you will be looking at this code later when you start interfacing with the robot. For now, keep scrolling down to button3.

c. OK. Now we’ll actually build our adding machine. We need to accomplish the following:

   Get the text from the two text fields
   Convert the text into numbers
   Add the numbers together
   Display the result

   Lucky for us, Java provides easy ways of doing all of this. Here is the code to accomplish the above steps. Type this into your file:

```
void button3_MouseClick(java.awt.event.MouseEvent event)
{
      int sum, add1, add2;
      String string1, string2;

      // Get the text from the text fields
      string1 = textField1.getText();
      string2 = textField2.getText();

      // Convert the text into numbers
      add1 = (Integer.parseInt(string1));
      add2 = (Integer.parseInt(string2));

      // Add the numbers together
      sum = add1 + add2;

      // Display the result in textField2
      textField2.setText(String.valueOf(sum));

      // Display the result in the command window
      System.out.println("The sum is " + String.valueOf(sum));
}
```

d. Now we’ll improve our interface a little by putting a more informative label on our button than ’button3’. Scroll up to the top of Robot.java and look for the declaration of button3 in the Robot() constructor. Find the line

```
      button3.setLabel("button3");
```

   and change it to

```
      button3.setLabel("Add");
```

   If you want to be even more fancy, you could change the starting text field entries:

```
      textField1.setText("Enter first number");
      textField1.setText("Enter second number");
```

e. Great! You’re done. Save your changes, go to the command window, compile the program, and run it:

```
      javac *.java
      java Robot
```

f. Can you guess what happens when you repeatedly hit the "add" button?

g. If you want, try making button4 do something interesting.

There you have it - an adding machine. But you also know how to (sort of) read and set textfields, react to button presses, compile java programs, etc.