

CSCI 161: Introduction to Programming I

Lab 1b: Hello, World (Eclipse, Java)

Goals

- to learn how to compile and execute a Java program

Overview

This activity will introduce you to the Java programming language. You will type in the Java program shown, compile it, and execute it. You will submit your project. Then, you will copy your program file, make some changes to it, compile it, and resubmit the project.

```
/* Hello.java: traditional first program
 * Author: your name
 * Date: today's date
 */
import java.util.Scanner;
import javax.swing.JOptionPane;

public class Hello {
    /**
     * @param args
     */

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        String personName;

        // Console input and output
        Scanner input = new Scanner(System.in);
        System.out.print("Please type your name: ");
        personName = input.nextLine();
        System.out.println("Hello, " + personName);

        // Graphical input and output
        personName = JOptionPane.showInputDialog("Please type your name: ");
        JOptionPane.showMessageDialog(null, "Hello, " + personName);
    }
}
```

Creating a Java program (Hello.java)

Open a terminal window and change your working directory to be your ~/161/Lab_1 directory (type **cd 161/Lab_1**). Now open Eclipse by typing '**eclipse &**' (**be careful not to omit the ampersand**). When the Eclipse workbench appears, your Lab_1 project will appear in the Package Explorer at the left of the window. Your myinfo.txt file will be open in the editor pane.

Create a new Java class file by clicking on the 'New Java Class' button (it is a green circle with a 'C' and a little +). If you click the arrow, you need to select 'Class' (the top entry). A new dialog box will appear. Leave the source folder as 'Lab_1' and leave the package blank. Type 'Hello' as the Name. Under 'Which method stubs would you like to create?', click on the box to create 'public static void main(String[] args)'. Now click the 'Finish' button.

In the editor pane, you will see a new tab entitled 'Hello.java.' This file automatically has focus. This file contains the basic outline or stub of a Java class called 'Hello' – the first line should read 'public class Hello {'. You will fill in this stub with the code shown above.

Begin typing in the Java program shown above. Include your name in the comments, along with today's date. Type the "import" statements before the "public class Hello {" line. As you type statements, pay close attention to capitalization (some letters must be capitalized) and spacing, both within a line and between lines. You want your code to look **exactly** like that shown above.

Get into the habit of indenting the program as shown above. The Eclipse editor makes it easy to use consistent indentation by automatically indenting the next line after the enter key is pressed. If you press 'CONTROL-SHIFT-F', Eclipse will automatically indent all of your code.

This program has minimal comments. We will expect more on later assignments. Comments and indenting are essential to the readability of your program.

Type in the program logic just below the '// TODO Auto-generated method stub' comment line. Note that Eclipse will attempt to assist you as you are typing in the code by suggesting possible completions to what you have typed so far. You may ignore these suggestions for now.

Click the 'Run' button (the green circle with the white triangle that looks like a play button). (If you click the down arrow, you will need to select 'Run...'). If Eclipse cannot automatically create a launch configuration, a dialog box will appear. Highlight 'Java Application' and click the 'new' button (a page with a plus sign... if you hover, it says 'New launch configuration'). 'Hello' is added below the 'Java Application' and it will be highlighted. The Name should default to 'Hello,' the project to 'Lab_1', and the Main class (which is the name of the class containing the method or function 'main') to 'Hello.'

Click the 'Run' button to launch your application! If you have not saved the Hello.java file, you will be prompted to save your files – you can click the box for 'Always save resources before launching' to avoid being prompted to save in the future.

This program combines 1) console input/output and 2) graphical components/GUI (graphical user interface). For the console input and output, the program asks the user to enter their name from the console (located at the bottom of the workbench window in the Console tab). After the user enters their name in the Console and hits ENTER, the program greets the user, saying 'Hello,' followed by the name that the user has entered.

The program then displays an input dialog box prompting the user to enter their name. After the user types a name and hits the 'OK' button (or hits ENTER), a message box appears, greeting the user and using the provided name.

Debugging your program: If Eclipse detects a mistake in your code as you are typing in your program, the line will be marked with a red wavy underscore. You should attempt to correct the error before running your program. If you do run a program and run-time errors occur, error messages will be shown in the Console pane. Continue to debug (correct) your program until you are able to run it without exceptions.

Submit your Project as Lab_1b: In the terminal, use **submit** to submit your updated project. Make sure that you are in your Lab_1 directory. The file being submitted should be Hello.java.