Introduction to Eclipse

ENGI 9859, Fall 2013

September 8, 2013

1 Installation

Skip this section if Eclipse is already installed and working.

This section applies to Windows 7, but the procedure should be similar for other OSs

1. Check that you have the Java Runtime Environment (JRE) installed. Look for a folder called `C:\Program Files (x86)\Java\jre6` or `C:\Program Files\Java\jre6` . If no such file is on your computer, you may need to install the JRE first.

2. Down-load *Eclipse IDE for Java Developers* from

   `http://www.eclipse.org/downloads/`

   Get the 32 bit version.

3. Unzip the `eclipse-java-indigo-win32.zip` to `C:\Program Files`. (On 64 bit Windows, the folder should be `C:\Program Files (x86)`.)

4. In Windows Explorer find the file

   `C:\Program Files\eclipse-java-indigo-win32\eclipse\eclipse.exe`

   or

   `C:\Program Files (x86)\eclipse-java-indigo-win32\eclipse\eclipse.exe`

   Right click on the file and select Pin to Taskbar and/or Pin to Start Menu.

2 A Java program

The following lab assumes you have Eclipse for Java Developers.

1. Start Eclipse

   - If you get a 'workspace not available' message, click OK
   - Set the workspace to
     - For Windows 7: `C:\Users\`*yourUserNameHere*`\workspace`
     - For Windows XP on a labnet computer: `H:\MyDocuments\workspace`

- For other operating systems, put the path of your home directory followed by `workspace`.

- Click OK.
- If you are asked about "Usage Data Upload" just click on "Turn UDC feature off" and then on "finish".
- Close the 'Welcome' window if there is one. (Click on the X which is on the 'Welcome' tab.)

2. Create a new project:

   - On the File menu select "New / Java Project".
   - Set "Project Name" to "Lab0"
   - Ensure that the "Execution environment JRE is "JavaSE-1.6" or a higher number
   - Select "Create separate folders for sources and class files"
   - Click Finish.
   - If you are asked "Do you want to open this perspective now?", click on Yes.
   - In the "Package Explorer" view you should see your project.

3. Create a class

   - Right-click on the Lab0 project in the Package Explorer
   - On the context menu select "New / Class".
   - Set 'Source folder' to Lab0/src
   - Set the 'Package' to hello.
   - Set the 'Name' to HelloWorld
   - Set Modifiers to public
   - Set Superclass to java.lang.Object
   - Check check-boxes 'public static void main(String [] args)' and 'Generate comments'
   - Click on Finish.
   - HelloWorld.java should appear both in the Package Explorer and as a new tab in the editing area.

4. Add a methods to the class

```
static String constructGreeting( String name ) {
    return "Hello " + "Fred" + "!" ;
}

static void greetTheUser( Reader reader, PrintWriter writer) {
    writer.println( "What is your name?" ) ;
    writer.flush() ;
    Scanner scanner = new Scanner(reader) ;
    String name = scanner.next() ;
    String greeting = constructGreeting( name ) ;
```

```
        writer.println(greeting) ;
        writer.flush() ;
    }
```

5. At this point you should see certain words underlined in red and two small red boxes with white Xs at the left of the editor pane. These marks indicate errors in your code.

   - Left-click on the first red box. Double-click on "Import 'PrintWriter' (java.io)"
   - If you look at the top of the file, you will see that an import directive has been added to the file.
   - Similarly fix the remaining 2 errors.

6. Modify main

   - Change the main method to look like this

```
public static void main(String[] args) {
    Reader r = new InputStreamReader(System.in) ;
    PrintWriter w = new PrintWriter( new OutputStreamWriter( System.out )) ;
    greetTheUser( r, w ) ;
}
```

   - Eliminate the new errors by importing InputStreamReader and OutputStreamWriter from java.io.
   - Fix indentation by typing Ctrl-a and then Ctrl-i .
   - Save the file with Ctrl-s

7. Run the program:

   - In the Package Explorer, right click on the file 'HelloWorld.java'.
   - On the context menu select 'Run As / Java Application'
   - On the Console view, you should see the text "`What is your name?`"
   - Notice that above the Console view there is a red square. This indicates that the console is connected to a process (i.e. an execution of a program) and may be used to terminate that process.
   - Click on the Console view so that it obtains 'keyboard focus'.
   - Type your name and press the Enter key.
   - The process will now terminate. (The red square will turn grey to indicate that process is terminated.)
   - Click on the double X icon above the console to clear the console.

3 Adding some tests

1. Add JUnit to your build path
   - In the Package Explorer, right-click on the project (Lab0).
   - Select 'Properties'. A Dialog entitled 'Properties for Lab0' should appear.
   - Select 'Java Build Path' in the left-hand pane.
   - Select the 'Libraries' tab in the right-hand pane
   - Click on 'Add Library'
   - Select 'JUnit' and click on Next
   - Select 'JUnit 4' and click on Finish.
   - Click on OK to dismiss the properties page.
2. Create a JUnit test class.
   - In the Package Explorer, right-click on the project (Lab0).
   - On the context menu select 'New / Source folder'.
   - Call the folder 'testsrc' and click on Finish.
   - In the Package Explorer, right-click on 'testsrc'.
   - On the context menu select 'New / JUnit Test Case'
   - Fill out the form
     - Select 'New JUnit 4 test'
     - Source folder; Lab0/testsrc
     - Package: hello
     - Name: TestHelloWorld
     - Leave everything else blank.
   - Click on Finish
3. Add a test method
   - Edit the TestHelloWorld class to look like this:

```
package hello;

import static org.junit.Assert.* ;

public class TestHelloWorld {

    @Test public void testConstructGreeting() {
        String name = "Bob" ;
        String expected = "Hello Bob!" ;
        String actual = HelloWorld.constructGreeting(name) ;
        assertEquals( expected, actual ) ;
    }

}
```

- Save

4. Run your test method

   - In the package explorer, right-click on 'TestHelloWorld.java'
   - Select 'Run as / JUnit Test'
   - Note the red bar and that there is 1 Failure.
   - Double click on the top line of the 'Failure Trace'.
   - Click OK.
   - Fix the error in constructGreeting. Save.
   - Run the test again.

5. Add another test method.

   - This one should read

```
@Test public void testGreetTheUser() {
    Reader reader = new StringReader( "Vladamir" ) ;
    StringWriter writer = new StringWriter() ;
    HelloWorld.greetTheUser(reader, new PrintWriter(writer)) ;
    // String.format replace %n with the appropriate sequence of
    // characters representing the end of a line.
    String expected = String.format("What is your name?%nHello Vladamir!%n");
    String actual = writer.toString() ;
    assertEquals(expected, actual) ;
}
```

   - Add import declarations as needed and run your code. (All classes to import are in java.io.)

6. Run the tests again. There should be 2 runs, 0 errors and 0 failures.

4 Debugging

1. Set a breakpoint

   - Select the editor tab for HelloWorld.java
   - On the left and right side you will see a grey margin
   - Double click on the margin to the left of the "greetTheUser(r, w) ;"
   - You should see a small blue circle in the left margin. This indicates a breakpoint.
   - Double click on the breakpoint to remove the breakpoint.
   - Double click on the same spot to recreate the breakpoint.

2. Run the debugger

   - In the Package Explorer, right click on HelloWorld.java

- Select Debug as / Java Application
- If asked to confirm a perspective switch, click Yes.
- You program should run to the breakpoint and stop.

3. Play with the Debugger

   - You can inspect variables in the Variables view.
   - You can change breakpoints in the Breakpoints view.
   - In the Debug View (or Run menu), try Step Into, Step Over, Run to Line, Terminate, etc.

4. Return to the Java perspective

   - When you are done debugging, click on the "Java" perspective icon, which will be near the top right of the Eclipse window.

5 Finish

Save all files. From the File menu select Exit.

Back up the workspace to the university network.

- If you are on a labnet machine, simply copy the workspace to the H: drive if it is not already there.
- Otherwise, copy the workspace to your labnet home directory using an SCP client such as winSCP or MuCommander. The host is garfield.cs.mun.ca. The username and password are you labnet user name and password.

It is good to be in the habit of frequently backing up all important work to a remote location.

6 More

There is much much more to learn about Eclipse. It has many navigation, searching, and browsing features. It supports changes to your code such as renaming variables, methods, and classes. There are also many “plug-ins” that can enhance Eclipse with new sets of features.