Laboratory 1: Getting Started with Eclipse JDT

Name   : _____

Student Number : _____

Laboratory Time : _____

Objectives

- Know your Windows and UNIX IDs and passwords in ICS computer labs
- Anatomy of the Eclipse workbench
- Use Eclipse to write a small Java program
- Use Eclipse to refactor a program

Grading Checklist

By the end of the laboratory session, you need to demonstrate to the TA that you can do the following tasks. The TA will check off the items below that you have completed and collect this cover page from you.

- ☐ Customize new file comment
- ☐ Write a Java program using the JDT
- ☐ Create a new project from existing source code
- ☐ Perform a simple refactoring to the created project

TA Initials: _____

Instructions for the Laboratory

By the end of the laboratory, you will need to complete the tasks listed below. You will need to do some preparation in advance to ensure that you can complete all of them in the time available. When you have completed all of the tasks, show the TA your work to receive your grade. It's best if you complete all of the tasks and show your results to the TA all at once.

**Task 1: Log in to ICS machines**

In order to complete laboratory exercises, you will need to be able to access the computers in the ICS labs. If you have not used the lab computer before you will need to have an account created and passwords set up by ICS Computer Support. The lab assistant in ICS 364 or the offices nearby can help with this.

a) Know your ICS user ID
b) Know your ICS Windows password
c) Know your ICS UNIX password
 •   These two passwords are sometimes, but not always, the same. If you are setting up your account, making the passwords the same will make this simpler.
d) Find an application on a Windows machine that will let you log on to a UNIX machine (For example, SecureCRT).

**Task 2: Customize your Eclipse start up**

Eclipse is installed on the lab machines, so that the default settings are restored on a daily basis. This means if you make any changes to the preferences, they will be lost. This will be very annoying on subsequent laboratories and assignments. Fortunately, there is a workaround. For this task, you will customize an Eclipse shortcut and change the default comment that is generated for new Java files.

a) Find Eclipse on the Start Menu.
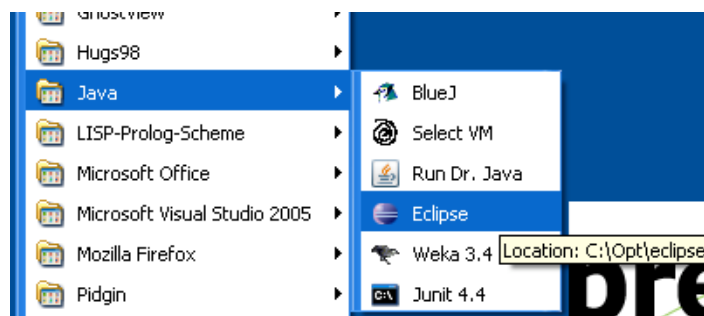
b) Make a shortcut to Eclipse on the desktop (or anywhere else that is convenient for you).

c) Go to the Properties for the shortcut by clicking on the icon with the right mouse button. Modify the "Start in:" box to start in your H: or home directory. This is where your Eclipse Preferences will be stored. Below, you see C:\Opt\eclipse – change this to, say, H:\Eclipse.
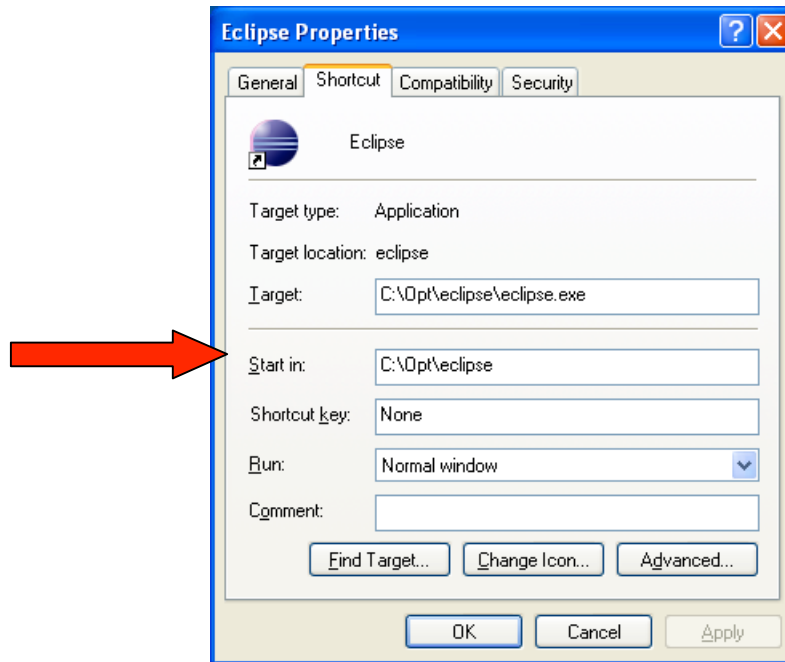

d) When you start Eclipse for the first time, the Launcher will ask you where you want to store your projects. Change this to a "workspace" directory on your H: or home directory. Make this the default, so it won't ask you again.
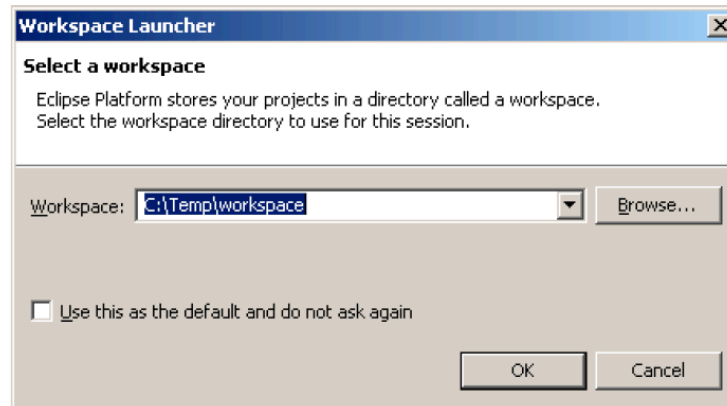
e) You will see the Welcome Window. Select the Workbench option.
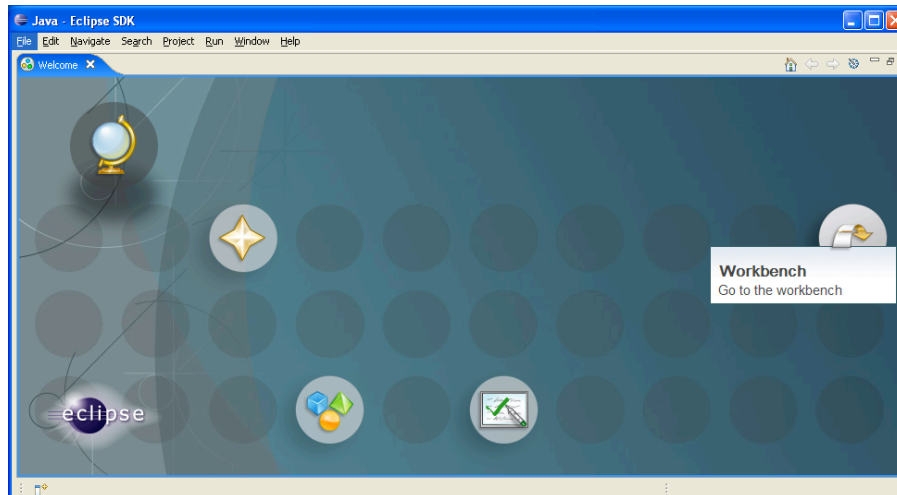
f) Modify the automatically generated comment for new Java files, so it includes your name, the course, the lab information, the university, and the copyright year.
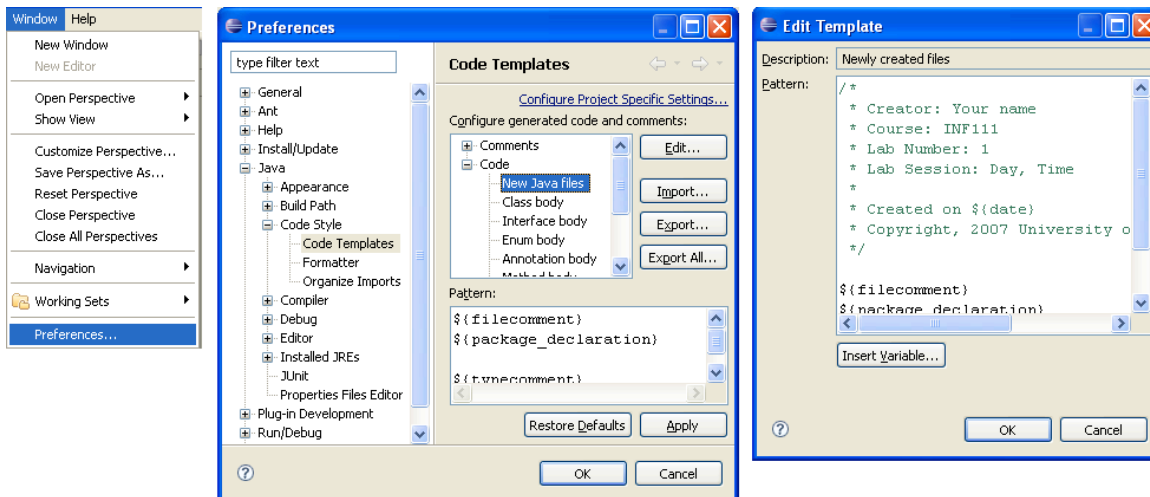
Task 3: Write a Java program using Eclipse JDT

Write a Java program that will print out your name and your expected grade in this course. The two should appear on a single line separated by a tab.

In order to do this, you will need to use the following features in the JDT.
   a) Create a new Java project
   b) Create a new Java package
   c) Create a new Java class

There are a number of ways to do this—the menu bar (File -> New), the context menu (right mouse button), and the Button bar. To run your program, go to Run -> Run As -> Java Application.

As you are working with the JDT, you should notice the kinds of automation that is build in to the tool. What kinds of warnings do you get about the names for the project, package, and class?

Also, take a look at Content Assist, which provides hints about the Java API.

If you need additional assistance, see the Help menu -> Help Contents.
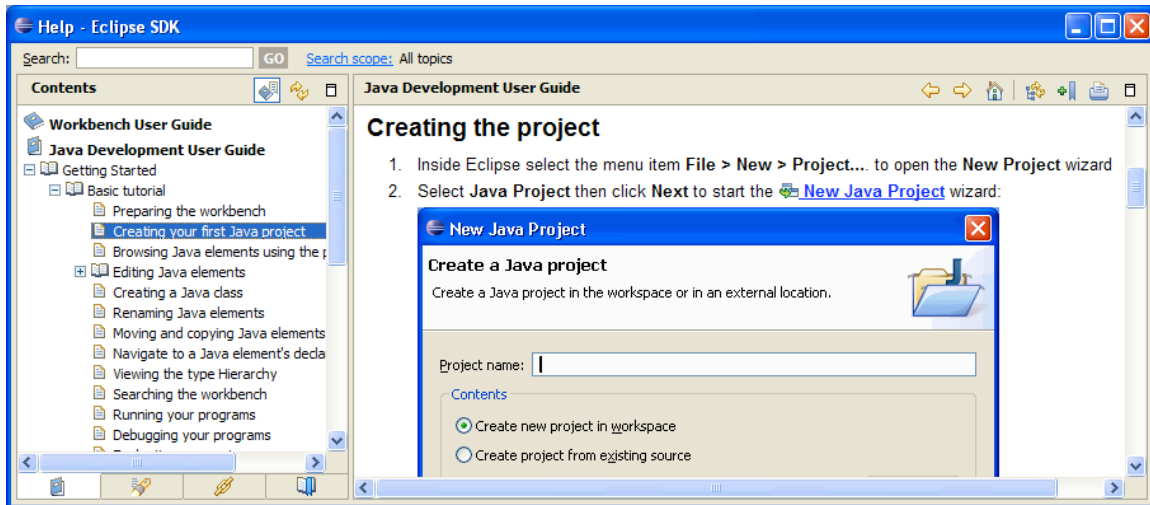
Grading: Show the TA the output from your program.

Task 4: Create a new project from existing source code
Start by unzipping the file provided to you on the webpage, in your Eclipse home directory. You should have the following files:

- Clock.java
- GUIEnvironment.java
- IObserver.java
- ISubject.java
- ObserverLL.java
- Processor.java
- SpaceCraft.java
- img/lunarlander.png

In Eclipse:

a) Create another Java project as you did in the previous task. Give it the name Lab1 and select the radio button "create project from existing source". Browse to your home directory and select the Finish button.
b) Check that the Project -> Build Automatically menu item is selected.
c) Click on the problems tab in the bottom right panel. You can see that there are 2 warnings generated. These could indicate that some variables or packages are not used. This is not the sort of warnings you should be bothered by. The TA will explain to you the difference between warnings, critical warnings and errors.
d) Click on the GUIEnvironment.java file, which contains the only main() method, and select Run -> Run As -> Java Application.

Grading: Show the TA that you created a new project from existing source code.

Task 5: Do a simple Refactoring

Now that the application is running, take a look at some of Eclipse's refactoring features.

a) Open the GUIEnvironment.java file and select the lines between Comment 1 and Comment 2. Right click on these highlighted lines and select Refactor -> Extract Method. Fill in the method name printInstructions in the dialog box and select the OK button. The highlighted lines are replaced with a call to the newly created method printInstructions(g2) and the highlighted lines make up the body of this new method.

b) Select all the java files for the Lab1 project (in default package) in the Package Explorer View and select Refactor -> Move. In the Move window, click on the Create Package button. Fill in the name for the package: edu.uci.ics.inf111.lunarlander and select the Finish button. Select the package edu.uci.ics.inf111.lunarlander and select the OK button. You will see that all the java classes belong to the created package.

c) Select the GUIEnvironment.java file in the Package Explorer View and select Refactor -> Rename. Fill in a new name of the Java element in the Rename dialog box and select the Finish button.

Grading: Show the TA your refactored code.