Introduction to Eclipse

Eclipse uses two different concepts for organizing code: projects and workspaces.

A project is a group of Java files that together form a single software application. If you want to create multiple separate software applications then you need to have multiple separate projects, with one project for each software application. A project also contains all the configuration information that Eclipse needs to compile and execute the corresponding software application. In Eclipse, all Java programming has to take place within a project. When you create a new project, Eclipse automatically creates a folder on the hard disk of the computer, which is going to contain all the information for the project, including all your Java code.

A workspace is then simply a folder on your computer hard disk inside which are stored multiple projects. In that folder Eclipse also stores some extra configuration information that Eclipse needs to keep track of the different projects you are working on.

To simplify your work, I advise you to create a new workspace folder for each new lab during the semester. Then, inside the workspace for a given lab, I advise you to create a new project for each lab question. For example, the Java code for question 7 of lab number 3 should be inside a project called "**`Question7`**" inside a workspace called "**`Lab3`**".

If you want to use a USB thumb drive to save all your work, I advise you to create the workspace folder on the desktop of the computer, and to copy the workspace folder from the computer's desktop to your USB thumb drive at the end of the lab. USB thumb drives are much slower than computer hard disks, so trying to use the workspace folder directly from the USB thumb drive will make Eclipse very slow!

Start Eclipse, using Microsoft Windows's Start menu.

After starting Eclipse, a window opens that asks you which folder to use as a workspace. Click on the Browse button. In the new window, select the desktop of the computer and click OK. Back in the first window, add the name of the current lab to the name of the workspace, with a \ between "**`Desktop`**" and the name of the current lab. For example, if you are currently doing lab number 3, then the name of the workspace should look like this:

```
C:\Users\Name Of User\Desktop\Lab3
```

Then click on OK. This will create a new folder called Lab3 on the desktop of the computer and all your Java projects for lab 3 will later be inside this folder. At the end of the lab you can copy this folder from the desktop of the computer to a USB thumb drive if you want to keep a copy of all your Java code.

Once Eclipse is running, close the "Welcome" window by clicking on the X button next to the "Welcome" name.

You then need to create a project using the File → New → Java Project menu. This opens a window asking for the name of the project you want to create. Use the name of the current lab question. For example, if you are currently doing question 7 of the lab, then the name of the project should be **`Question7`**. Then click on Finish. After you have done this, your new project will appear in the left window of Eclipse.

Once you have created the project, use the File → New → Class menu to add a new Java class to the project. This opens a window asking for the name of the class you want to create. Use the name that is specified in the lab question you are currently working on. For this introduction to Eclipse, you can use the name **`MyClass`**. Then click on Finish. After you have done this, the new Java class is added to the project in the left Eclipse window and a text editor is opened in the middle Eclipse window for the file **`MyClass.java`** where you can start typing the code for the class.

To test Eclipse, copy-paste the following Java code into the editor for **`MyClass.java`**:

```
public class MyClass {
      public static void main(String[] args) {
            System.out.println("Hello World!");
      }
}
```

Once you have created this Java source file, you normally need to compile it into a bytecode program that can be run by the Java Virtual Machine. Fortunately for you, this is done automatically by Eclipse as you type the program.

If any code is underlined in red in Eclipse then it means that you have a mistake in your program. Move your mouse over the code underlined in red and Eclipse will then give you an error message describing the problem (and maybe also some suggestions about how to fix the problem). You then need to change your Java code to fix the mistake.

Some error messages might also appear in the Problems window at the bottom of Eclipse. If you double-click on one of these error message, Eclipse then highlights the piece of Java code where it thinks the mistake is. Again you then need to change your Java code to fix the mistake.

Once your Java code compiles correctly without any error or warning, then you are ready to run the bytecode on the Java Virtual Machine. To do this, simply click in the text editor with the right mouse button and use the Run As → Java Application menu. A new window will open, asking you which "resources" (files) to save. Select "Always save resources before launching" and click on OK. Your program will then run. You can see the output of your program inside the Console window at the bottom of Eclipse.

Every time you make a change to the Java code of your software, Eclipse will automatically compile your code into bytecode and, if there are no errors, you can then execute the program.

If you have a Java project that contains multiple Java classes, then the Run As → Java Application menu will only be available when you click with the right mouse button in a text editor for a Java file that contains a main method. You can also find the same menu by clicking with the right mouse button on the name of the project. Eclipse will then let you select which Java file with a main method you want to execute.

When you are finished with a project and want to start a new one (for example, when you have finished answering a lab question and want to start working on the next lab question), simply create a new project. Both the old project and the new project will appear in Eclipse and be saved in the same workspace. This makes it easy to copy Java code between different projects (for example, between different questions for the same lab). You can also directly copy-paste an entire existing project to create a new project which is a copy of the previous one.

A few more details:

- If you want to change to another workspace (for example, to look at the Java code from an earlier lab), you can use the File → Switch Workspace menu.
- The Help → Help Contents menu opens a new window that has a complete explanation of how to use Eclipse. See the Workbench User guide and the Java development user guide.