

1 A Quick Introduction to the Eclipse IDE

Eclipse is an integrated development environment (IDE) for Java programming. Actually, it is capable of much more than just compiling Java programs but that is primarily what we will be using it for. Eclipse is a professional development environment used by programmers in numerous different fields from mobile app development to medical imaging to rocket guidance systems. Eclipse has a ton of features, in this course we will be using only a small fraction of the options available in Eclipse. This lab is designed to get you started with the basics.

1.1 Creating a Workspace

Before we open up Eclipse we want to create a place to store our programs. Open up Windows Explorer, navigate to your P drive and create a new folder called COSC117, this is where you will store all of your programs for this class. I would also suggest storing copies of all your programming work on a flash drive or up in the cloud.

Now launch Eclipse by double clicking the eclipse icon in the Academic Software window. When you do a Workspace Launcher will appear. YOUR WORKSPACE IS VERY IMPORTANT! This is where Eclipse will find all of your programs. If you select the wrong workspace you may not find your files or even worse they could be deleted by others, if your workspace is saved to a shared location on the computer.

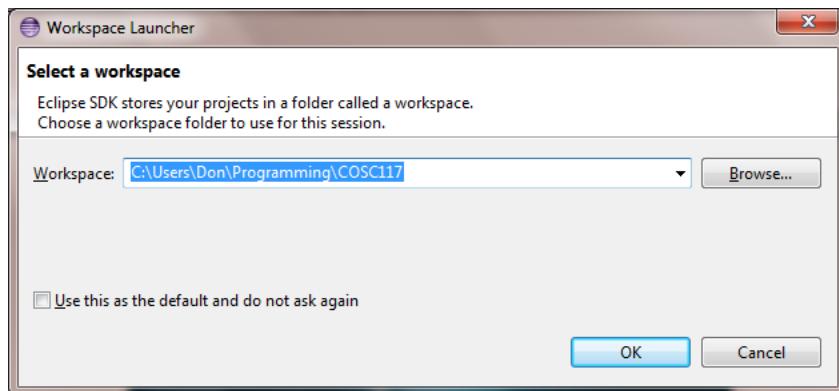


Figure 1: Workspace Selector

Click on the Browse... button and select P:\COSC117 (the folder you created above) and then click OK. Leave the “Use this as the default and do not ask again” check box unchecked. At this point you should see the following.

Click the X on the Welcome tab to close the welcome window. Now you should see the Main IDE Setup.

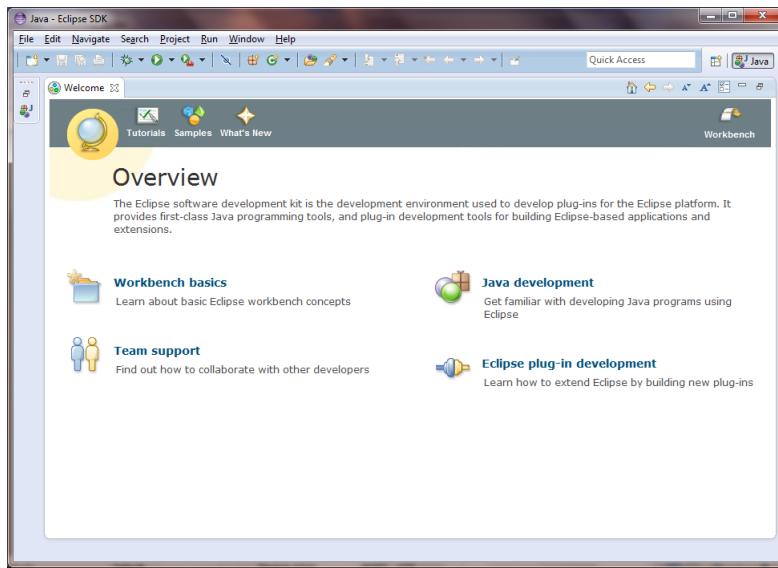


Figure 2: Welcome Screen

1.2 The Eclipse IDE

There are four main areas in the interface.

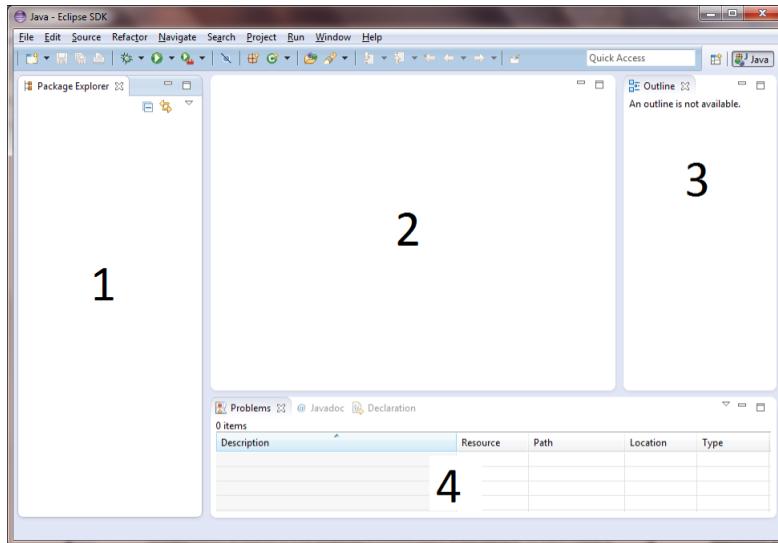


Figure 3: Main IDE Setup

1. Package Explorer: This area will hold all of the projects (that is, programs) you write and keep in that workspace. Professional programmers will organize their work by using multiple workspaces but for this class a single workspace holding all of our programs will be sufficient.
2. Editing Area: This area is where you will write the code for your programs.

3. Outline: This area will contain a sort of outline to your program. That is, it will list all of the “functions” contained in your program and make navigation of your program much easier. This area may not seem too useful at the start when our programs are small but as they get bigger and more complex using the outline area will save you a lot of time.
4. Output: The output area has many functions. It is where you will see descriptions of your errors and where you will see “console” output of your programs.

2 Your First Java Program: Hello World

To create a Java program in Eclipse we need to first create a Java Project. To do this select File > New > Java Project from the main menu, the far left tool in the toolbar is the New tool so clicking this and selecting Java Project from the drop-down menu will also work. When you do either of these, the New Java Project dialog will appear.

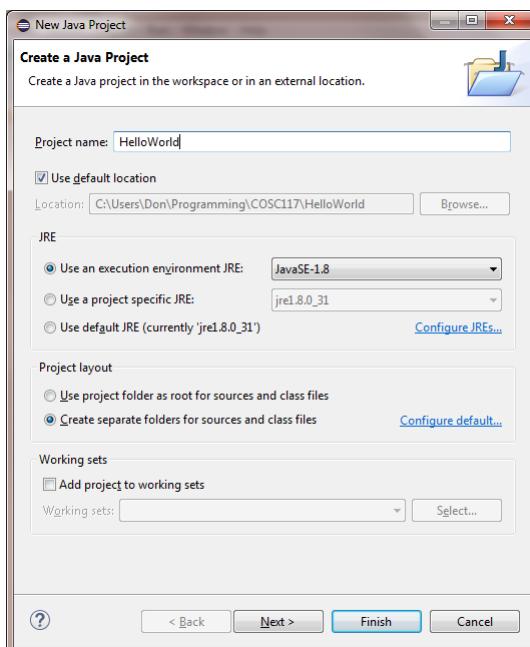


Figure 4: Project Creation

Each project must have a name, type in `HelloWorld` and then click Finish. All of the other default settings will be fine. At this point the Package Explorer window will display the new project `HelloWorld`. If you click the little triangle beside the `HelloWorld` title you will see several sub folders of information. One of these is `src`, which stands for source, this is where all of the Java code files will be stored for this project. Each project has its own set of code files. **So each programming exercise will have its own project** and for most of the semester each project will have a single code file in it. Notice that the `src` folder is empty at this point. This is because we need to add a code file to the project.

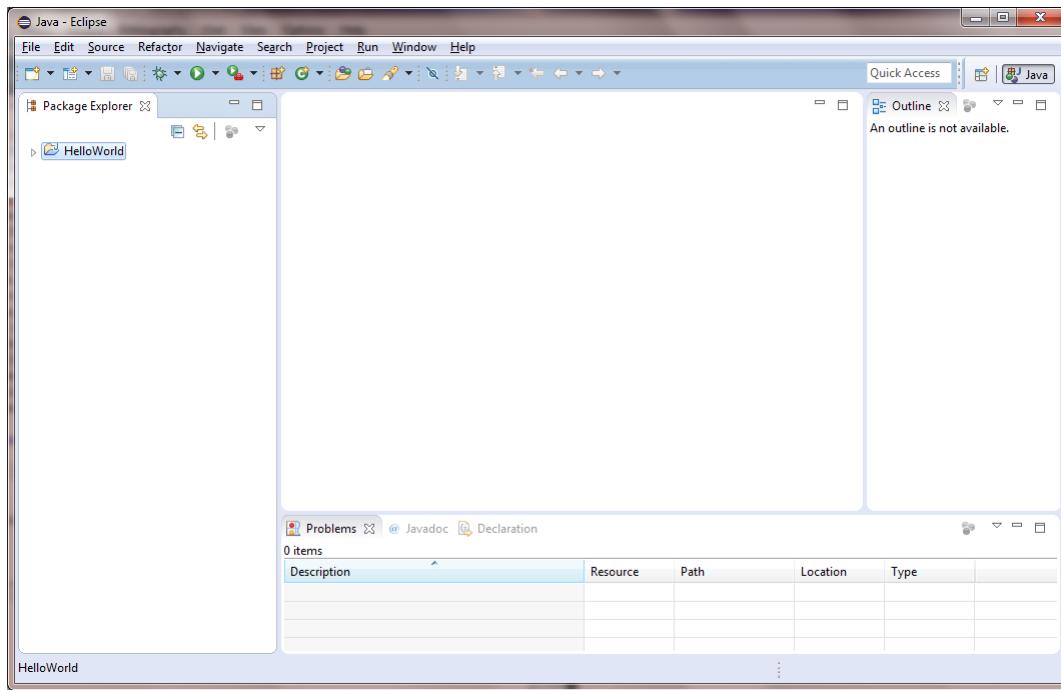


Figure 5: Workspace with HelloWorld

To add a code file select File > New > Class from the main menu, the far left tool in the toolbar is the New tool so clicking this and selecting Class from the drop-down menu will also work. When you do either of these, the New Class dialog will appear.

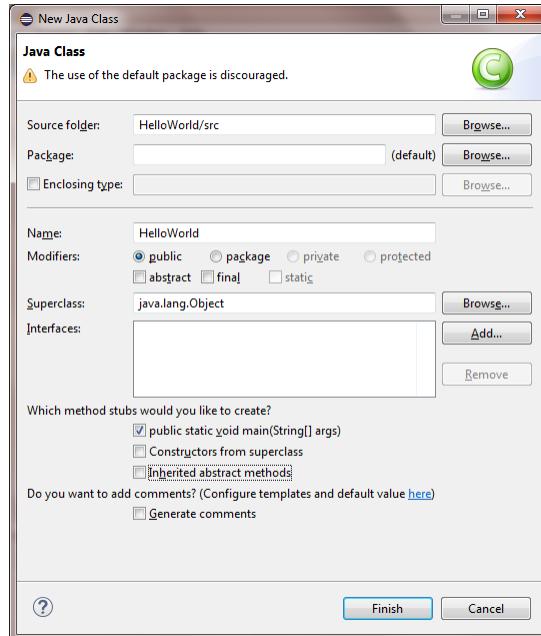


Figure 6: Java File Creation

You must give the class a name, it is not necessary to give the class the same name as

the project but for now it is convenient to do so. So type in the name `HelloWorld`. Also, select the `public static void main(String[] args)` check box and uncheck all of the others, leave the other options as they are and click Finish. At this point there will be one file in the `src` folder and Eclipse will put in a template structure into the editing window, as seen below.

```
1 public class HelloWorld {  
2  
3     public static void main(String[] args) {  
4         // TODO Auto-generated method stub  
5     }  
6  
7 }  
8  
9 }
```

Do not worry about what all the words are, we will discuss their meaning as the semester goes on. For now, edit the code so that the editing window looks like the following. Of course, you should put your name in as the author in place of mine and today's date for the date.

```
1 public class HelloWorld {  
2  
3     /**  
4      * The Hello World program, in Java.  
5      * Author: Don Spickler  
6      * Date: 8/27/2018  
7      */  
8  
9     public static void main(String[] args) {  
10        System.out.println("Hello World");  
11    }  
12 }
```

Now we are ready to run the program. Select `Run > Run` from the main menu or click the run tool button, the big green play button. At this point you should see `Hello World` appear in the console window at the bottom of the screen. If you get an error instead, check to make sure that your code looks like the code of the program above.

3 Submitting Your Work

You will normally submit your work through the MyClasses page for this class and usually it will simply be the `.java` file that you created. So you will need to know where these files are.

1. Your workspace is at `P:\COSC117`, so that is where you start. Open up Windows Explorer, navigate to your P drive and then the `COSC117` folder. You will see a couple folders that start with a dot, leave these alone, and the folder `HelloWorld`.
2. Go into `HelloWorld` and you will see folder `bin` and `src` and some other stuff.
3. Go into `src` and you should see `HelloWorld.java`, this is the file you created which has your code in it. You can open this file in Notepad or Notepad++ and you should see the actual code you wrote, do not edit the file outside of Eclipse.

In general, you would find your source files, usually there will be only one, in the src folder in the folder with the name of your project.

To submit this file to me for a grade, do the following.

1. Go to the MyClasses page for this class.
2. Select Assignments on the left.
3. Select Lab 1.
4. Click on the Submit Assignment button.
5. In the File Upload box, click the Browse... button.
6. Navigate to the HelloWorld.java file as you did above and select the file. At this point the HelloWorld.java name should be beside the Browse... button.
7. Click on the Submit Assignment button below. You are done. There should be a small submission box in the upper right that tells you that you turned the assignment in and when it was submitted.

4 Installing Eclipse on Your Personal Computer

If you want to install Eclipse on your personal computer, follow the instructions for your computer platform.

4.1 Windows

1. Download the Eclipse IDE from the Eclipse website.
 - (a) Go to eclipse.org.
 - (b) Click on the Download button.
 - (c) Click the Download Packages link under Get Eclipse.
 - (d) Click the Eclipse IDE for Java Developers link.
 - (e) Click either the Windows 32-bit or the Windows 64-bit download link depending on your system, the bit version of your JVM must match the bit version of Eclipse. You may wish to download them both if you do not know which one is correct for your computer. If you get an error when you try one of them the other should work.
 - (f) Click Download, the file will be stored in the Downloads folder of your computer. The file will be a zip file with a name that begins with eclipse-java.

2. Unzip the file. The process here will depend on the zip utility you have installed. In most cases you can right click the zip file, go to either the zip utility submenu or it may be listed in the main popup menu, and select something like Extract Here. When this is finished you should have a new folder called `eclipse`.
3. Most programs that are designed for Windows computers have an installation process, Eclipse does not, it is completely stand-alone, and is ready to run. You may move the `eclipse` folder to any place on your computer, just remember where it is since you will need to use it to start Eclipse.
4. After you have moved the folder to where you want it, go into the `eclipse` folder and you will see a file `eclipse.exe` with the Eclipse icon. If you double-click `eclipse.exe`, Eclipse should start. If you get an error, you may have downloaded the wrong bit version of Eclipse, if so, go back and download the other version and repeat the above process.
5. Once Eclipse has started you will be asked for the workspace you want to use. If you have not already created a folder for your work, you may wish to cancel Eclipse and create a folder before opening Eclipse. You are now ready to go. You can create a shortcut to Eclipse and place it on your desktop and/or pin it to your start menu or taskbar.
6. At this point you can delete the `eclipse-java-XXX.zip` file (where XXX is just the rest of the name).

4.2 Mac

1. Download the Eclipse IDE from the Eclipse website.
 - (a) Go to eclipse.org.
 - (b) Click on the Download button.
 - (c) Click the Download Packages link under Get Eclipse.
 - (d) Click the Eclipse IDE for Java Developers link.
 - (e) Click Mac OS X (Cocoa) 64-bit.
 - (f) Click Download, the file will be stored in the Downloads folder of your computer. The file will be a tar.gz file with a name that begins with `eclipse-java`.
2. The download will go to your Downloads folder. You will normally have an icon for the Downloads folder on the right-hand side of the dock. Find `eclipse-java-XXX.tar.gz` (where XXX is just the rest of the name) in your Downloads folder, and drag it to the Desktop. Then double-click it. You will see a folder named “`eclipse`.”
3. Drag the “`eclipse`” folder into your Applications folder. The easiest way to do so is to open a new window in the Finder and click on Applications in the list you get on the left-hand side. Then drag the “`eclipse`” folder in with the other applications. Make sure that you do not drag it into a folder that’s already within Applications. In other

words, when you're done, the Applications folder should have directly within it a folder named "eclipse."

4. (This step is not required, but it's strongly recommended.) Double click the "eclipse" folder. You'll see an application named "Eclipse"; it has a purple icon with white horizontal stripes. Drag it into your dock. Now you will be able to launch Eclipse by clicking on the icon in the dock.
5. You may now drag `eclipse-java-XXX.tar.gz` to the Trash. Empty the Trash whenever you wish.
6. When you launch Eclipse for the first time, you'll be asked "'Eclipse' is an application downloaded from the Internet. Are you sure you want to open it?" Click "Open."

Note that if you get an error that the version of Java is too old then you will need to update the Java JVM. This is common on the Mac. Search for "Java JDK" and you should get the correct site on the first hit or so. You will be taken to the Oracle web site where you can download the Java JDK. Follow the install instructions from the Oracle web site.

4.3 Linux

1. Go to eclipse.org.
2. Click on the Download button.
3. Click the Download Packages link under Get Eclipse.
4. Click the Eclipse IDE for Java Developers link.
5. Download either a 32 or 64 bit version, whichever matches your distribution.
6. The download will be a `.tar.gz` file, untar it, copy the `eclipse` folder to wherever you want it, and run the `eclipse` executable file in the `eclipse` folder.

Most likely, Eclipse will also be installable from your distribution's software manager. It might be an older version but that will not matter for us. In case your version of Java is too old, your software manager should have OpenJDK available for download.

5 Class Examples

On the MyClasses page for this class (or on the Classroom Resources page of my web site) there are links to the examples code for this class. This is a single zip file that contains a single Eclipse workspace with all of the example projects. Unzip the file and you should have a new folder named `AllJavaCodeExamples` that contains approximately 150 example projects. Copy this folder to your P drive and personal computer. To access the examples in Eclipse, just switch your workspace to the `AllJavaCodeExamples` folder.