Eclipse Installation Guide

Version 2

Table of Contents

Introduction

These Eclipse installation instructions are used by students enrolled in the Master of Science in Software Development and other Computer Science Department programs in both on-campus and online programs. The document begins with a discussion of Eclipse, downloading and installing Eclipse, creating a Hello World project, and exporting that project for submission. Note that as new versions of Eclipse are released, some of the screens may look different than the screenshots in this document. Nevertheless, this guide will help get you started quickly on most any version of Eclipse.

Although the examples in the main document are for the Microsoft Windows family, including Windows 8 and Windows 10, you may use Eclipse on Linux, a Mac, and many other platforms. You will need to follow a separate guide for downloading and installing Eclipse on those platforms, and you can skip down to the Creating a Hello World Project section after Eclipse is installed on your platform.

If you can't determine how to proceed or something goes wrong, and web searches don't help, ask your facilitator or instructor for help. Good luck, and have fun!

Eclipse Overview

Eclipse is one of the most popular Java application development environments for desktop and Android applications. Eclipse is cross-platform and runs under Windows, Linux, Mac OS, and other platforms. It supports advanced Java development from a default install, and can be further extended with a variety of plugins to suit particular environments. Eclipse supports basic and advanced debugging, as well as a host of application creation tools. By learning to use Eclipse, you'll have experience with an industry standard Java development environment.

Eclipse is entirely open source with many contributors to both its base code and to plugins. This feature makes it popular and extensible.

Downloading and Installing Eclipse

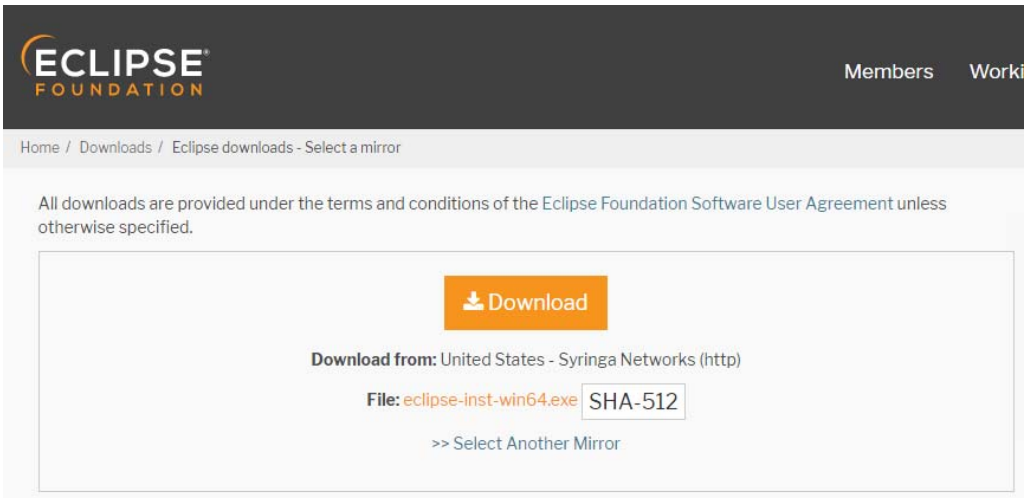
Step 1: Downloading Eclipse

Visit Website

Go to https://www.eclipse.org/downloads/packages/installer to get started downloading Eclipse. The Eclipse Foundation regularly updates their website, so what you see may be different than the following.

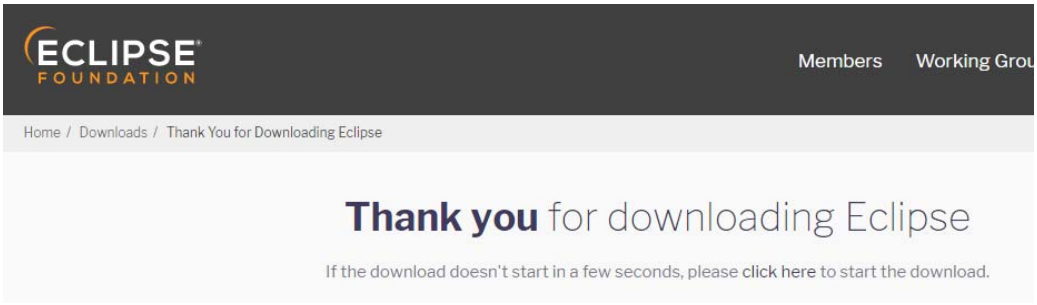
Click Platform Link

Click the "Windows 64 bit" link to get started. After doing so, you will see a screen similar to the following.



Click Download Link

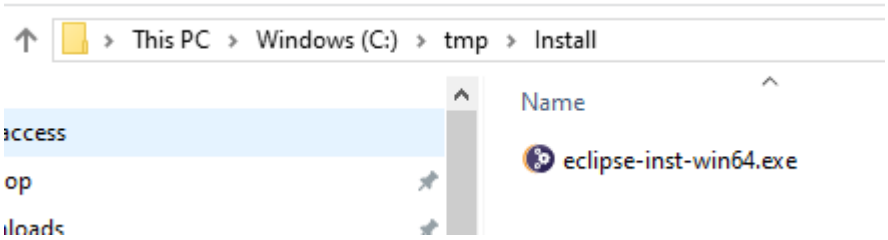
Click the “Download” button to continue. You will then see a screen like the following:



Your browser will save the executable, possibly asking you where you’d like to save it depending upon your browser settings.

Browse and Execute

Next, browse to where you downloaded your executable in Windows Explore. Alternatively, your browser may give you an option to run it directly, without the needed to navigate in Windows Explorer.

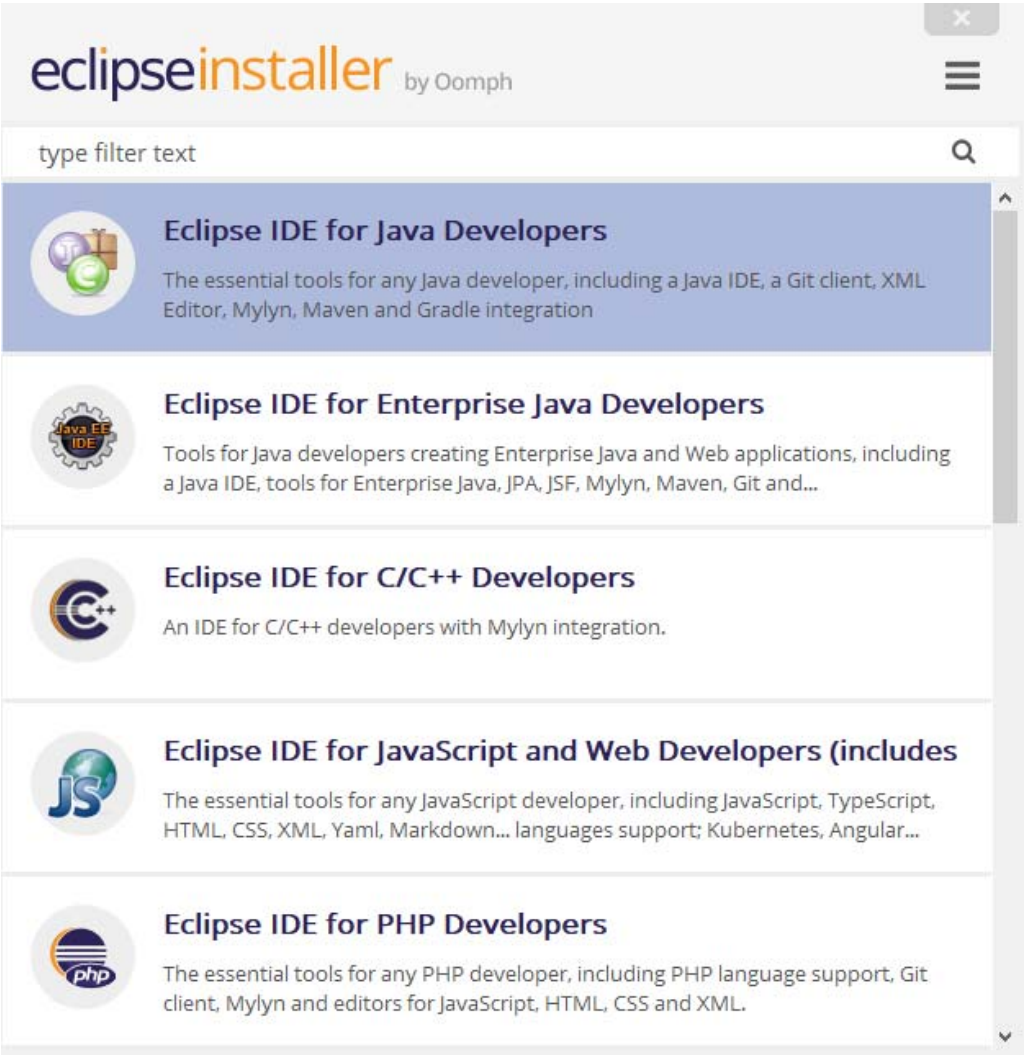


Execute the installer.

Step 2: Installing Eclipse

Select Flavor

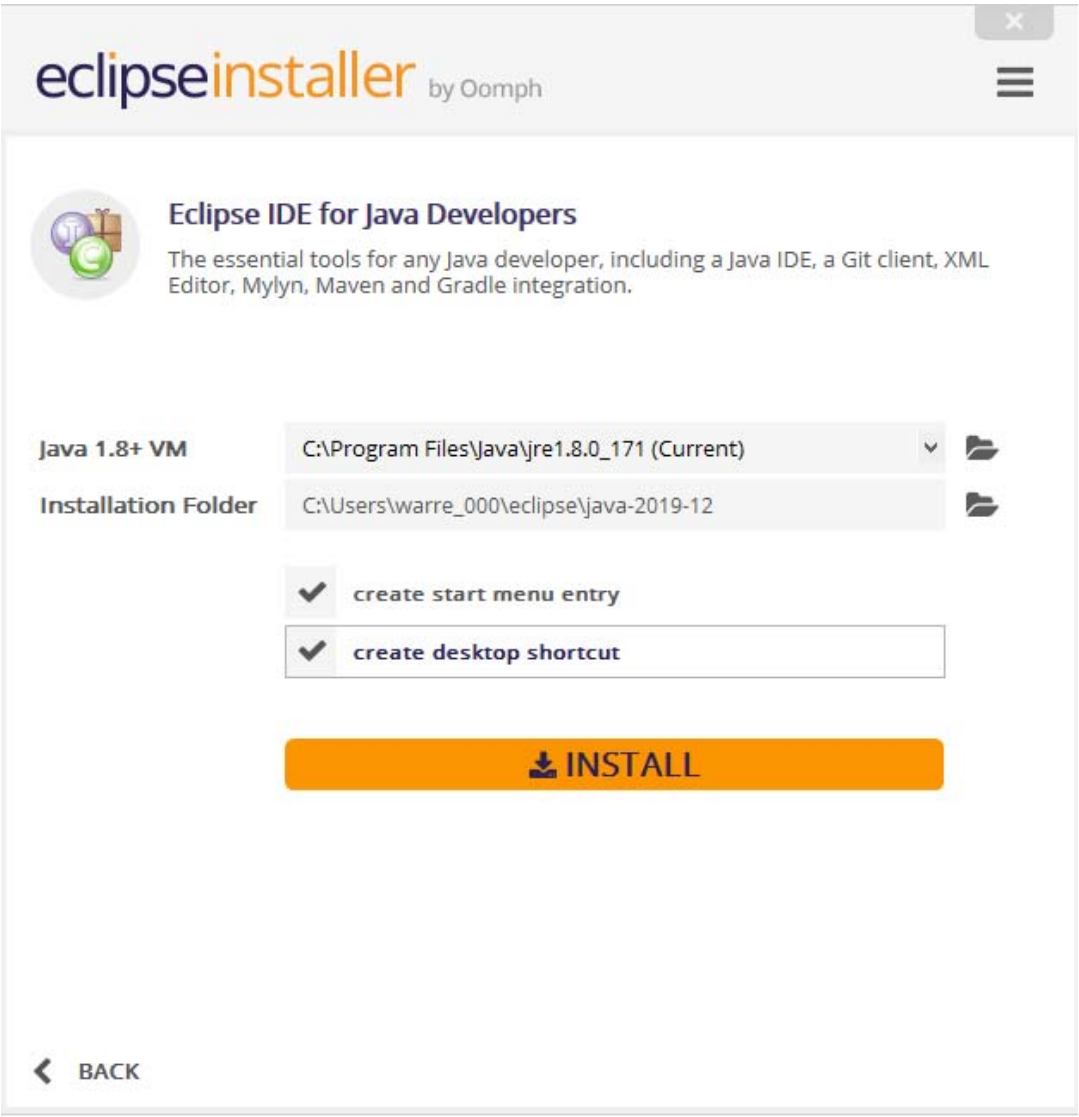
After executing the installer, you will see a screen like this.



Unless you have a good reason, select the “Eclipse IDE for Java Developers” flavor. It has everything you need to get started, and you can add additional plugins later, should you need. After doing so, you will see a screen similar to the following.

Choose Options

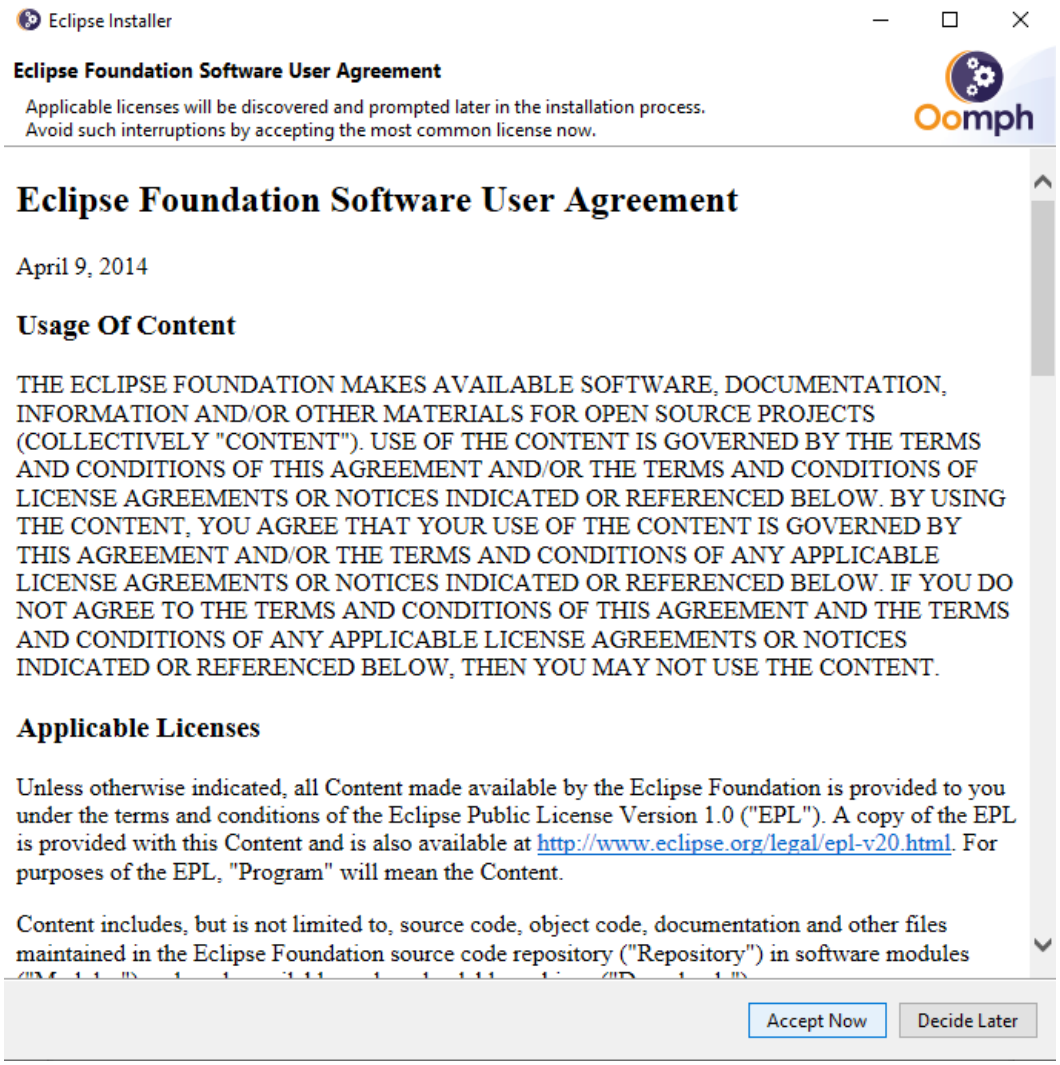
The next screen lets you choose a couple of options, which Java version to use, and what directory to install into.



Unless you have a good reason not too, accept the defaults and click the “Install” button.

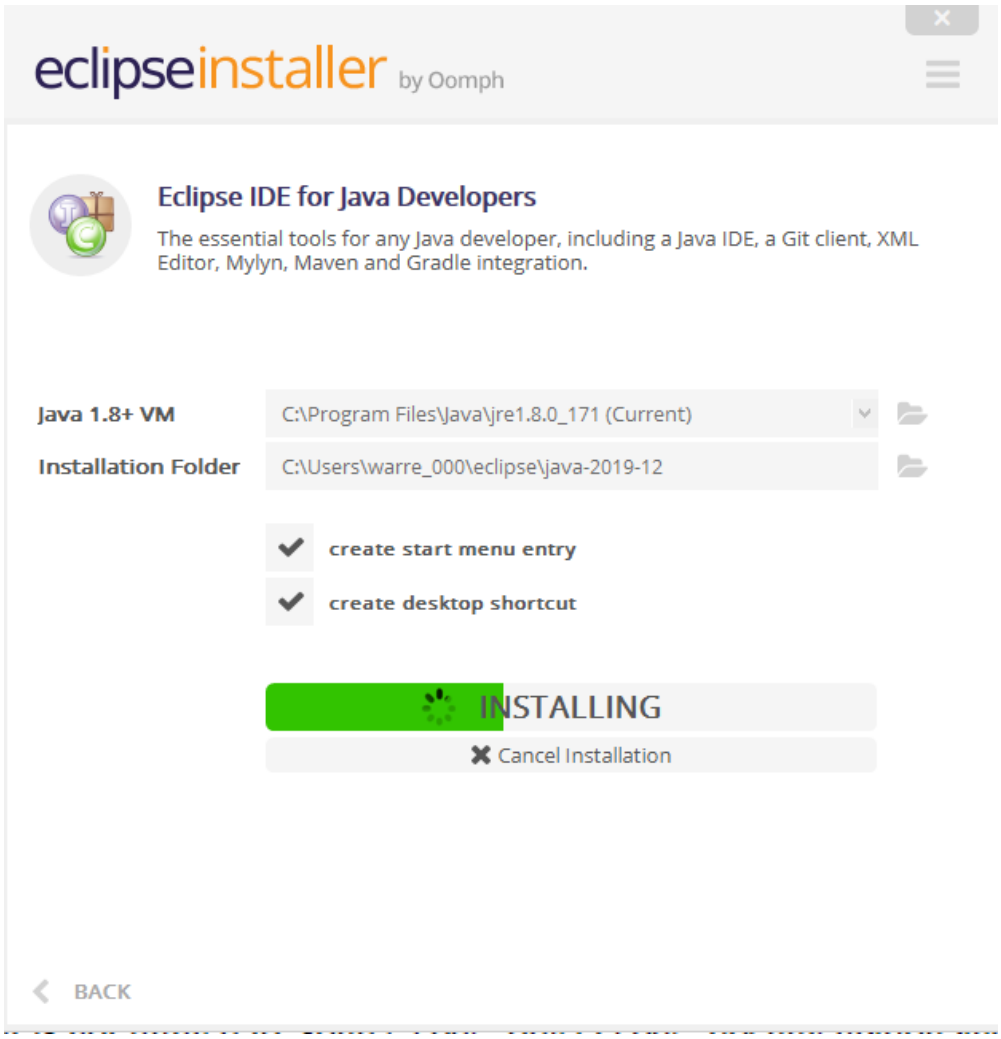
Accept the License

Before installation begins, you'll be asked to accept the license for Eclipse like the following screen shown below.



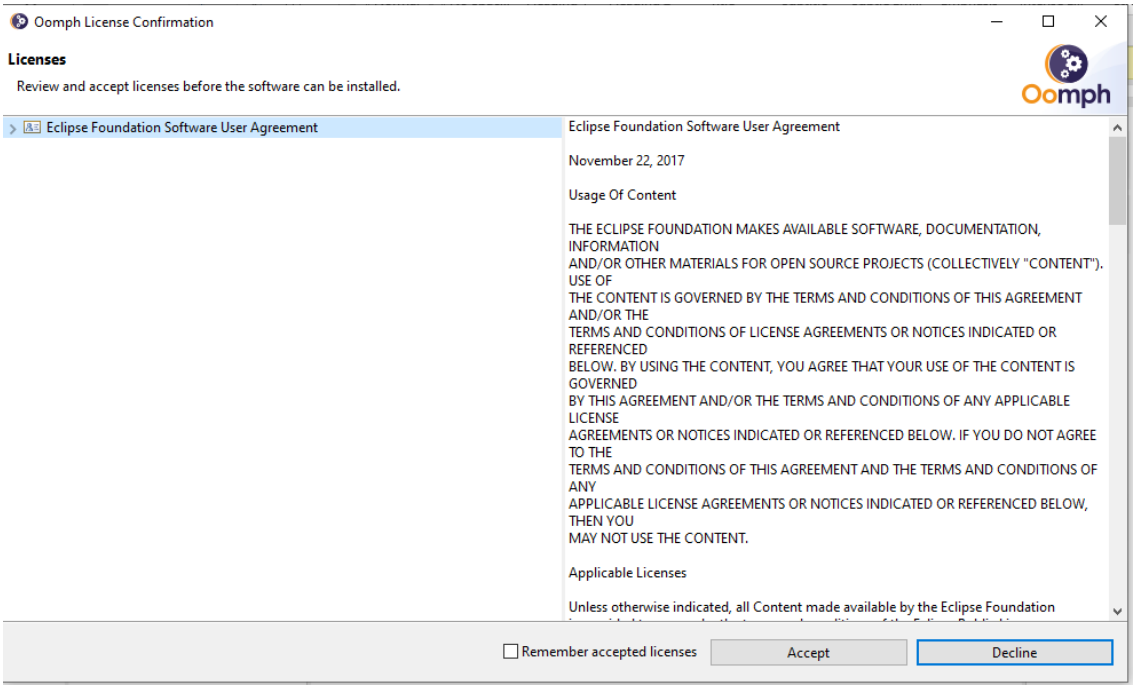
Start Installation

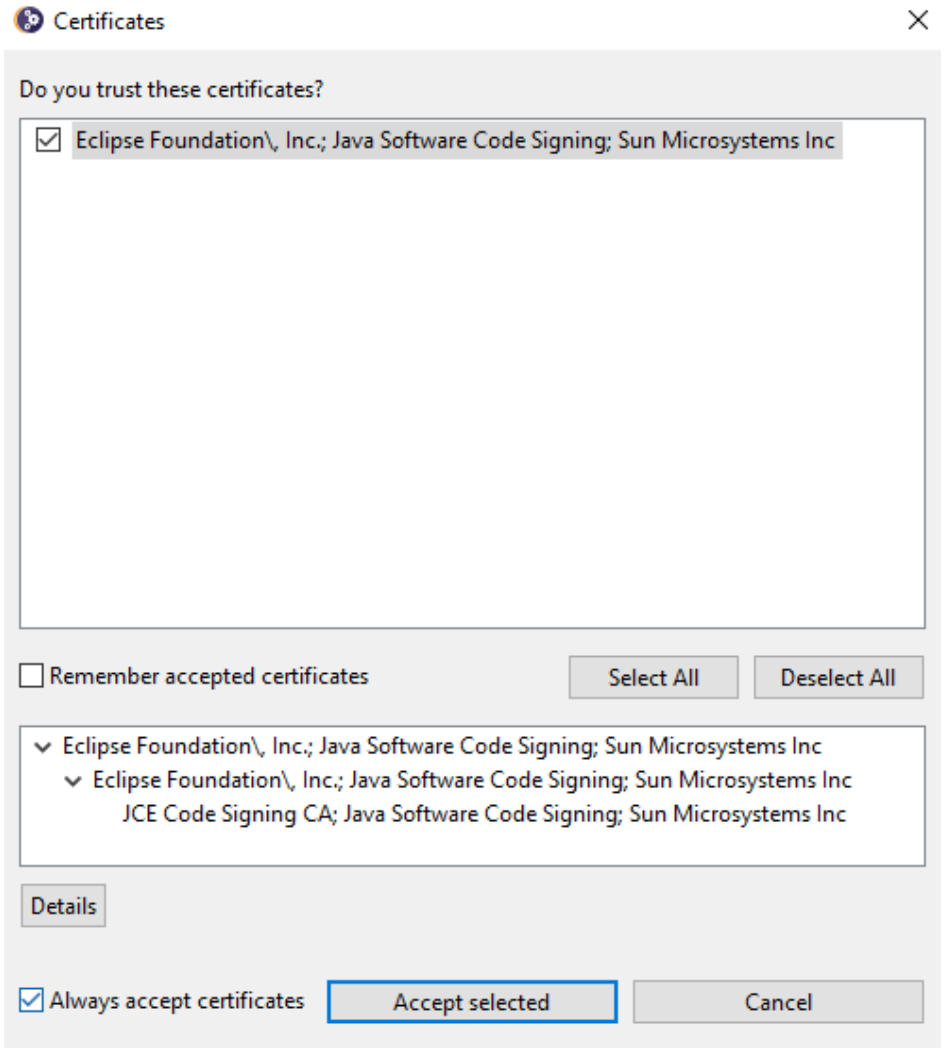
After accepting the license, Eclipse will begin installing and you will see a progress bar.



Additional Acceptance

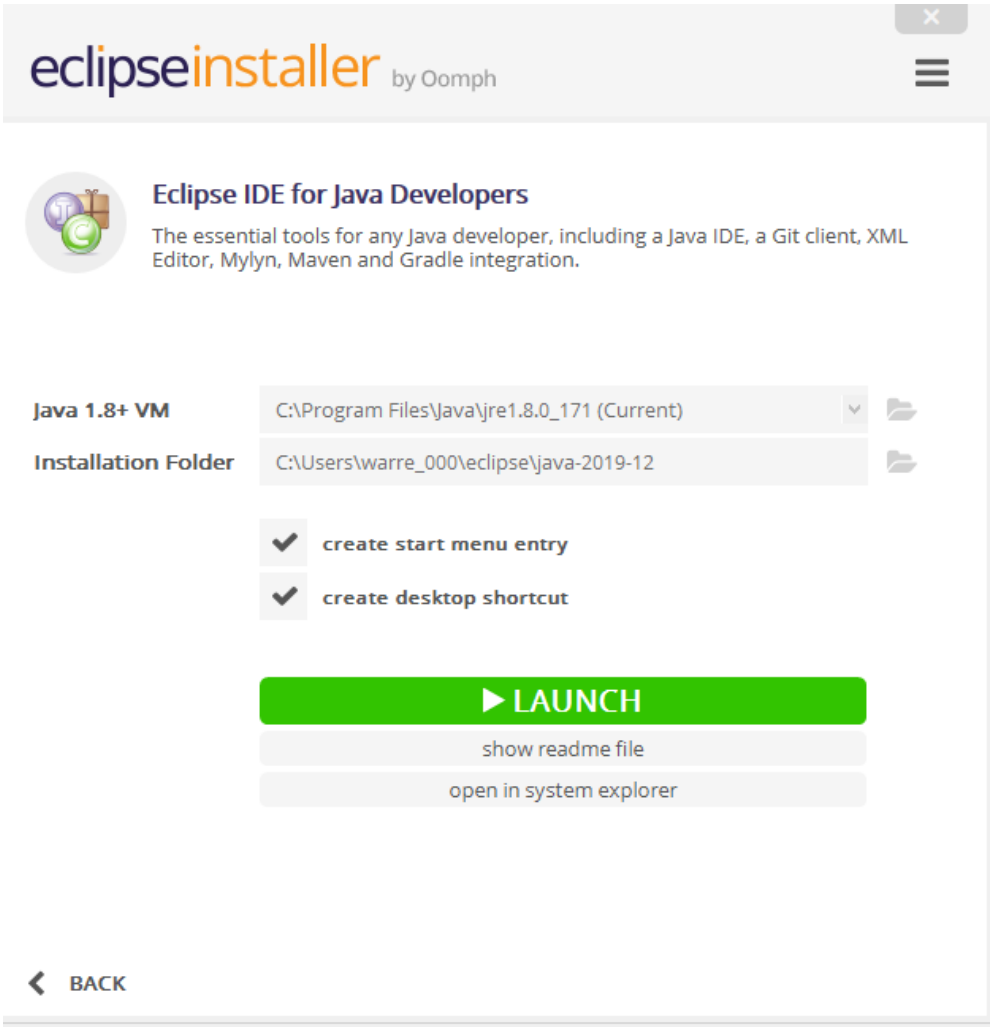
Note that a license agreement and certificate agreement will come up partway through the install, so if your install does not appear to be progressing, double check whether a second window has appeared asking for acceptance. The two screens look like the following.




Certificates
Do you trust these certificates?
Eclipse Foundation\, Inc.; Java Software Code Signing; Sun Microsystems Inc
Remember accepted certificates
Select All
Deselect All
Eclipse Foundation\, Inc.; Java Software Code Signing; Sun Microsystems Inc
Eclipse Foundation\, Inc.; Java Software Code Signing; Sun Microsystems Inc
JCE Code Signing CA; Java Software Code Signing; Sun Microsystems Inc
Details
Always accept certificates
Accept selected
Cancel

Install Completed

After the install has finished, you will see a screen similar to the following.



Congratulations! Eclipse is now installed on your machine and is ready to use.

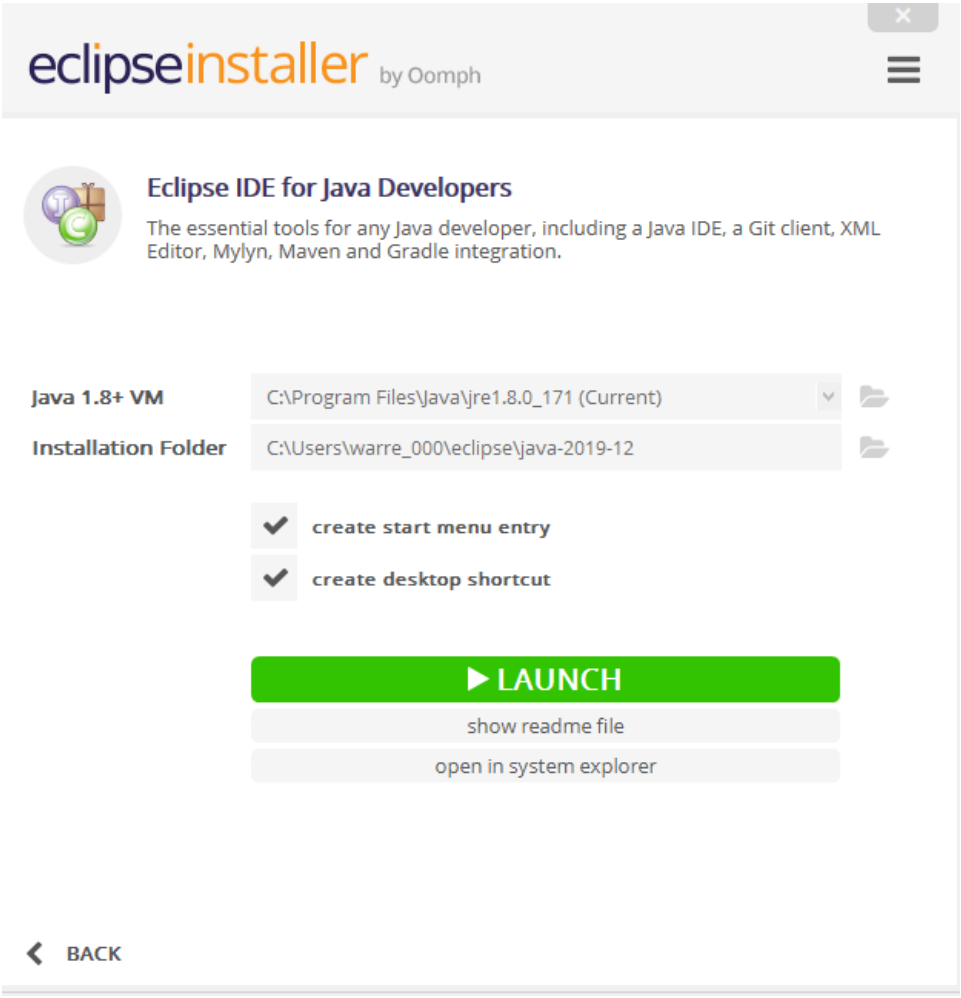
Creating a Hello World Project

To get you started, we'll walk you through creating your very first project. It will simply print "Hello World" to the screen.

Step 1: Launching Eclipse

Launching from Installer

After the installation is completed, you can launch Eclipse for the first time by clicking the "Launch" button.



Launch from Windows

Going forward, you can launch Eclipse from Windows, either from the Start Menu, or from the shortcut on the desktop.

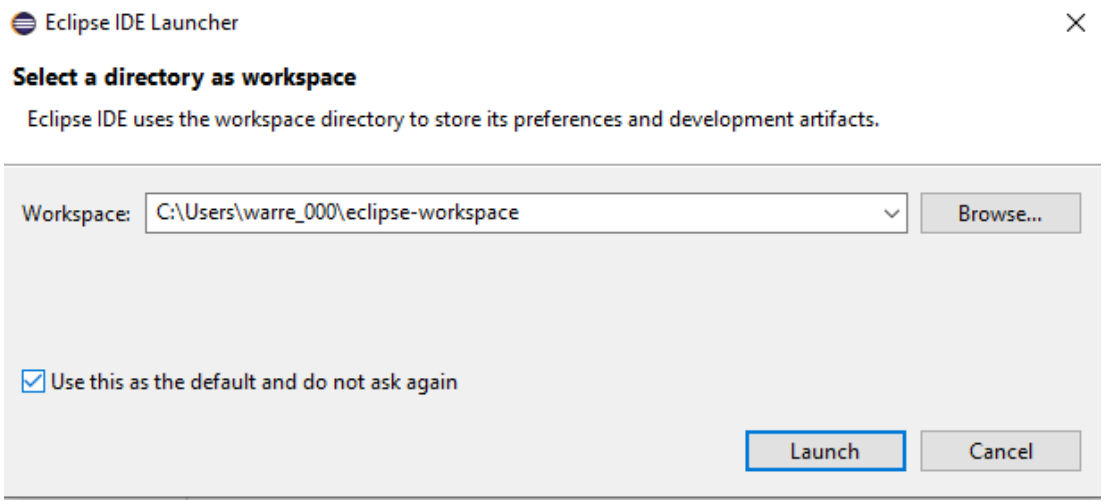
Start Menu

Desktop

Step 2: Choosing a Workspace

Workspace Selection

The first time Eclipse launches, you are asked you to decide on a base directory in which it will create the files it needs, including project files, classpath files, JARs, and many other types of files. You do not typically need to work with this directory directly, so it is fine to accept the default choice.

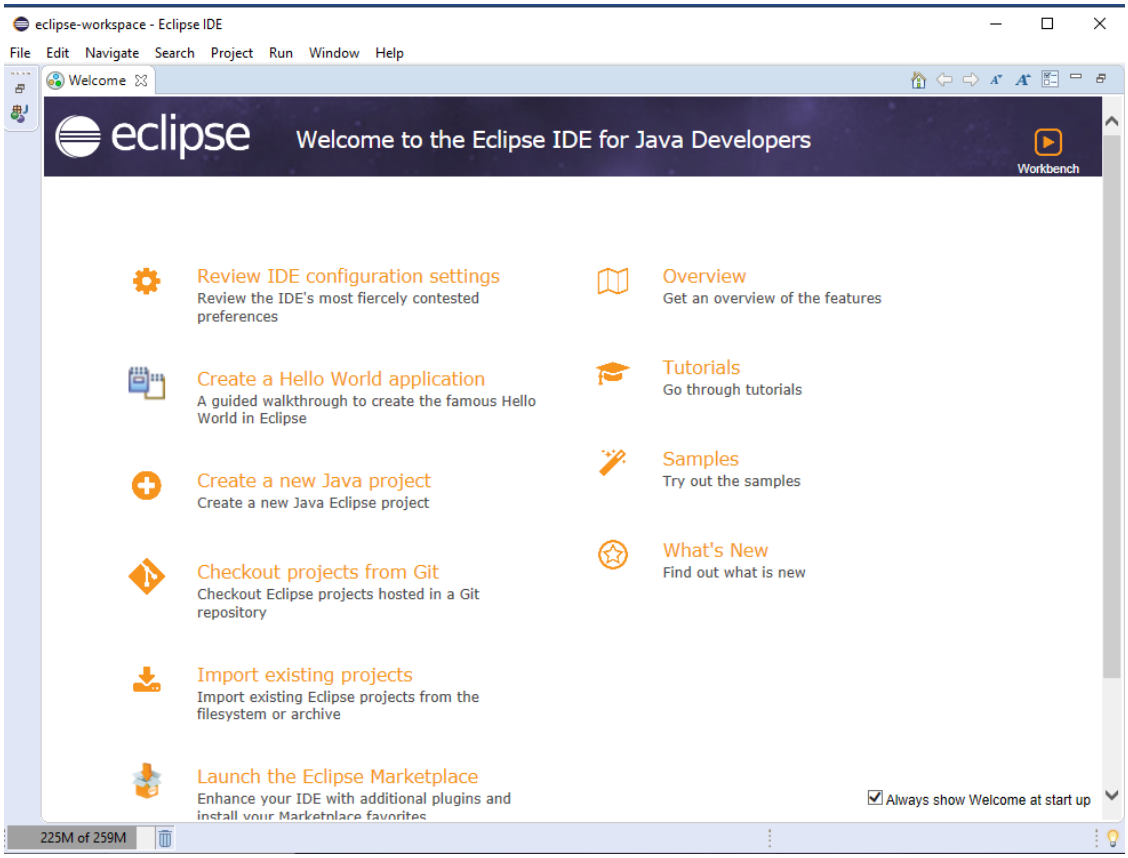


By default, Eclipse is configured your user's "User" directory, which is the preferred location on Windows for application files. Check the "Use this as the default and do not ask again" checkbox so that you don't need to reconfirm this every time you start Eclipse.

Step 3: Creating the Project

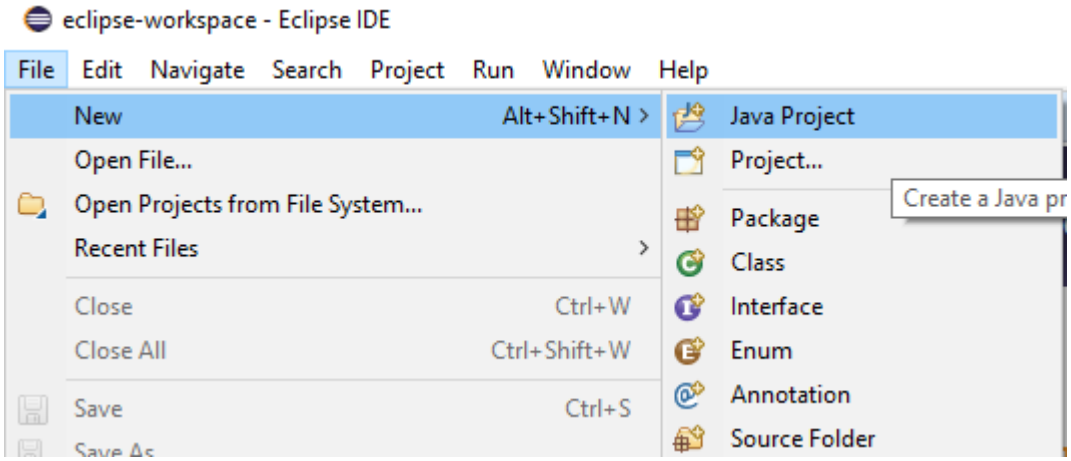
Initial Appearance

After clicking the “Launch” button, you’ll first see a splash screen with a progress bar, followed by the initial screen in Eclipse. The two screens are shown below.



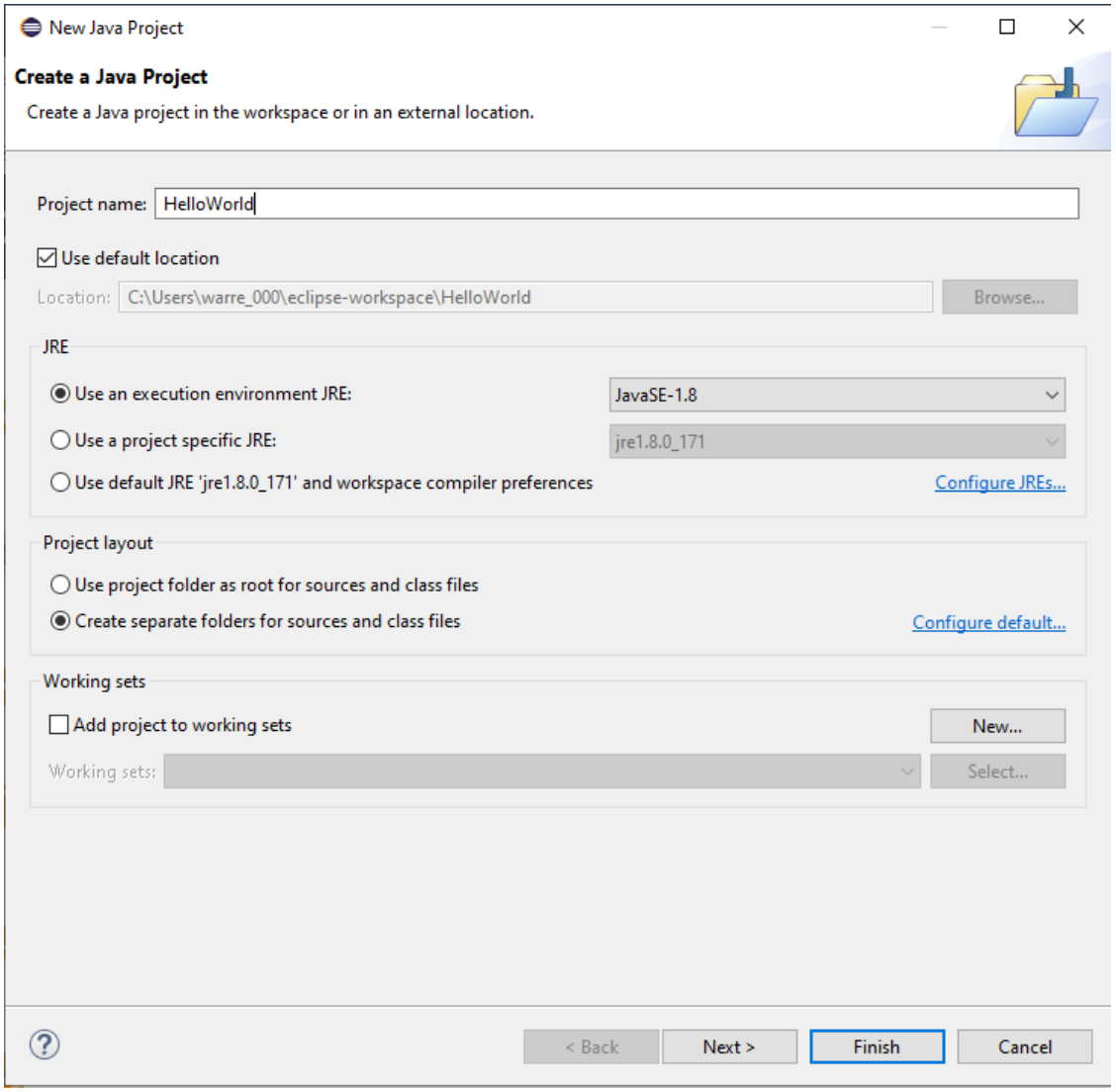
Create a Project

On the initial screen, you see a lot of menu options and links to click. Don't worry about the plethora of options now; you can learn those over time as you become more familiar with Eclipse. You only need one option to create a Java Project. Click the File/New/Java Project menu option to get started.



Name the Project

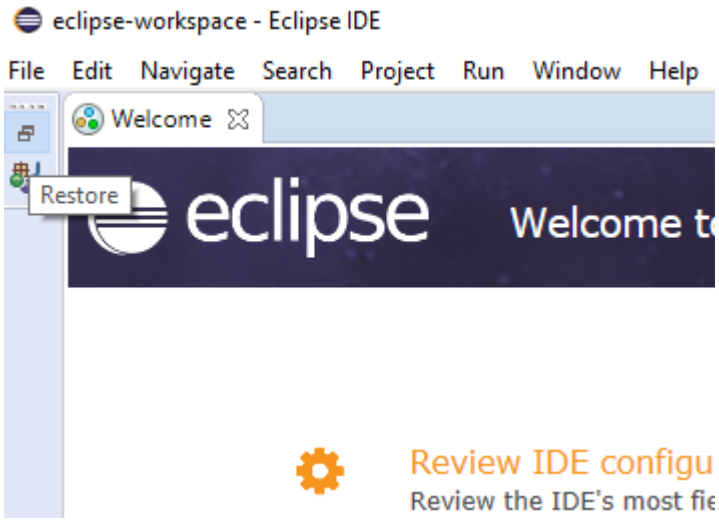
Next, you want to create a name for your project. Following Java convention, you should use camel case for the name. With camel case, spaces are not included in the name, and each word is capitalized. So, “Hello World” is “HelloWorld” in camel case. Type the name in the “Project Name” field.



Leave everything else with their default options, and click the “Finish” button.

Expand the Panes

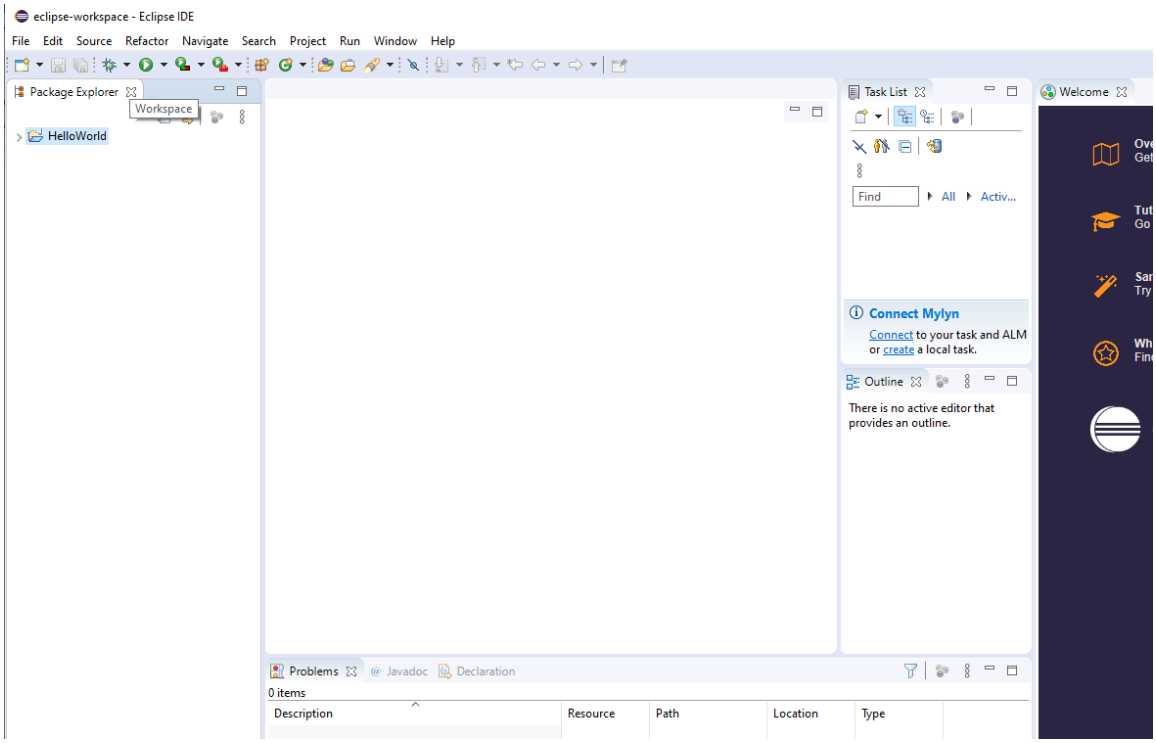
When the project is first opened, it almost looks like nothing has changed as you still see the welcome screen. In the upper left, however, there are panes that have been minimized by default.



Click the “Restore” option (which looks like two windows stacked on each other) to expand the panes.

Remove Unneeded Panes

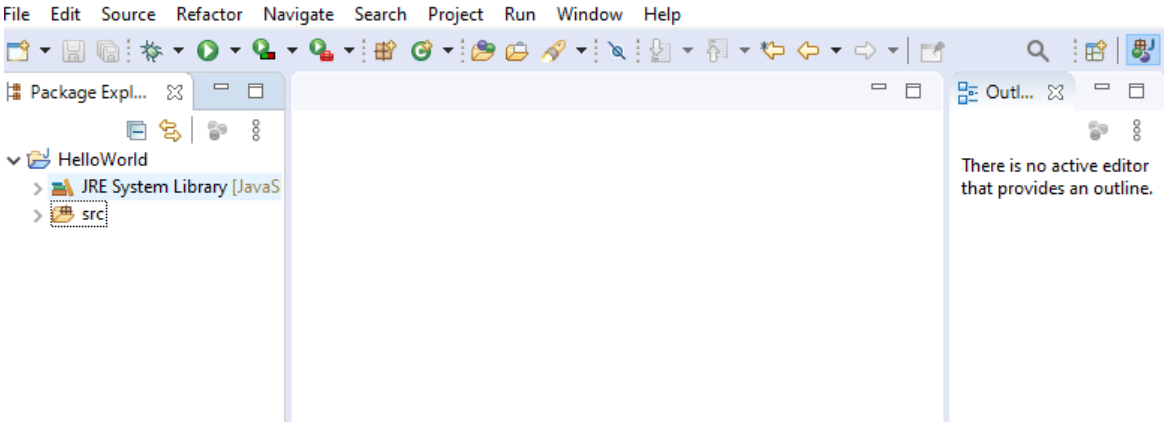
Once the panes have been restored, you will see the very useful “Package Explorer” on the left, the coding buffer in the middle, and a “Task List” and “Welcome” on the right, as follows.



Go ahead and click the “X” on the “Task List” and “Welcome” panes to close them because you don’t need them and they clutter up the space.

Understand The Workspace

With those panes removed, your workspace will now look like the below.



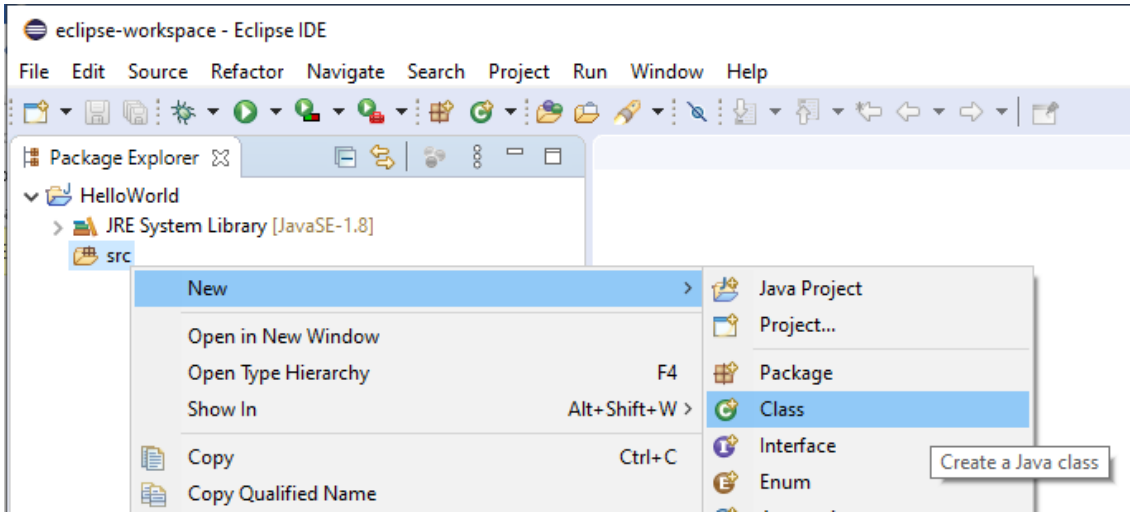
The “Package Explorer” pane gives you a high-level view of the libraries and source files you have in your project. Since you have not yet created any source files, you won’t see much here at the moment. Use the package explorer to navigate between source files as needed.

The middle pane is the buffer where you edit the code of the currently selected source file. Since you are not currently editing anything, it is blank.

The “Outline” pane will show you a class outline of the current class you are editing.

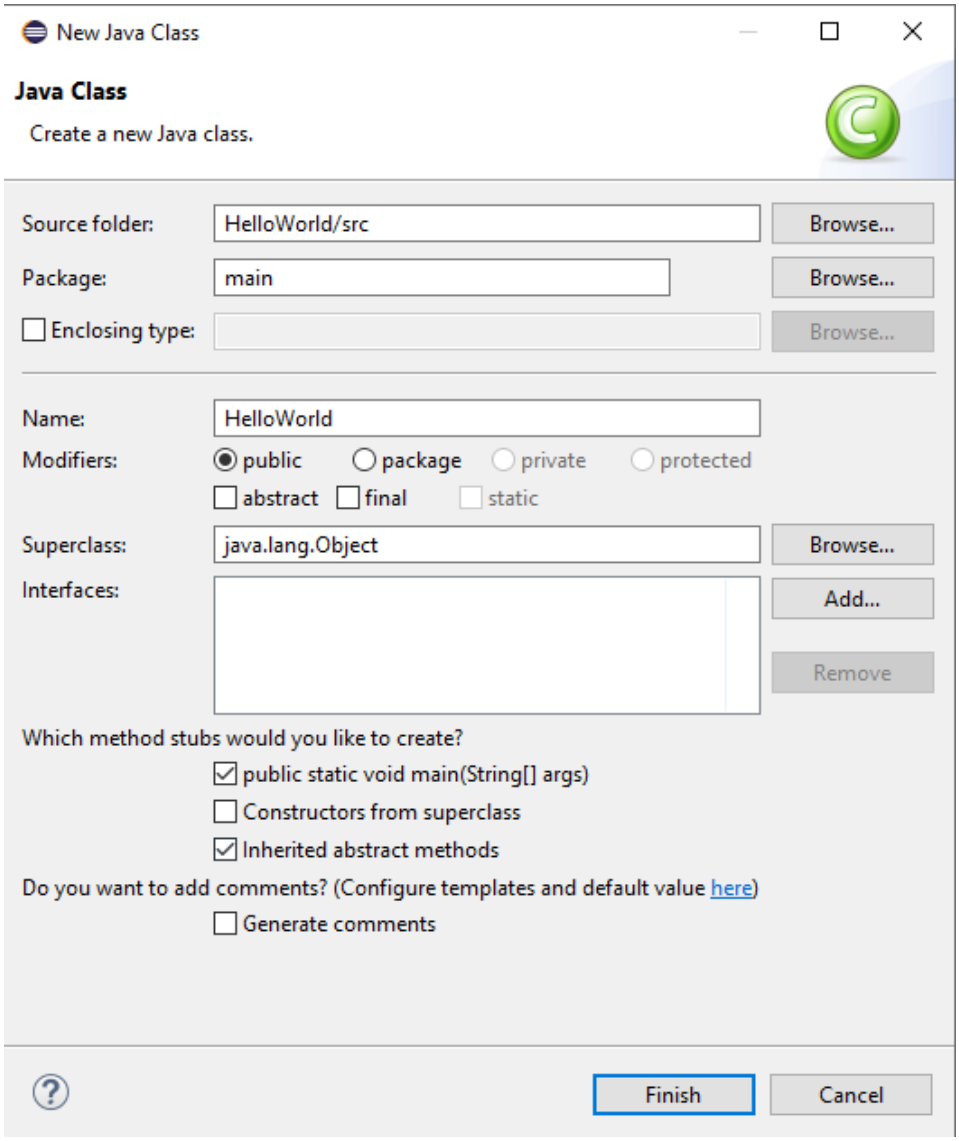
Create a File

The next step is to create your main Java class which will print out the “Hello World!” text. Right click on the “src” folder in the package explorer, and then click New/Class.



Choose Class Options

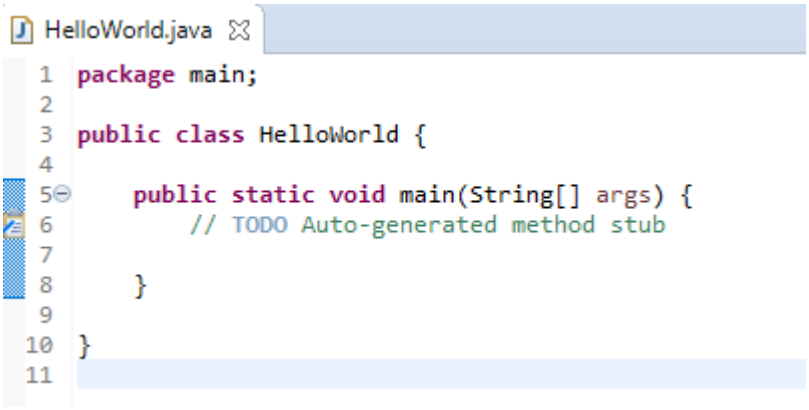
The next screen will ask you to name your class and select other options about your class. For your name, you will want to follow the Java convention of using camel case, and name your class “HelloWorld” in the “Name” field.



Since this will be our main class that should execute, select the “public static void main(String[] args)” checkbox to have it generate that method. The last option you need to change is the package. Using the default Java package is discouraged, so type a package name in the “Package” field. In this case, we used the name “main”. When creating larger projects, you may want to use a different package name as needed by your project.

Understand the Code

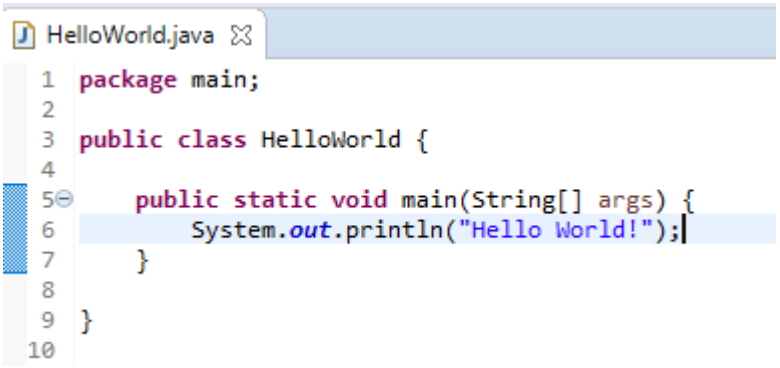
After clicking the “Finish” button on the class creation screen, the buffer pane shows the new HelloWorld class you created.



Notice that its in the “main” package, its name is “HelloWorld”, and it has a default main method.

Edit the Code

Now, you need to add a System.out.println() method call to print out “Hello World!” when the class is executed. To do so, type in the following line in the code.



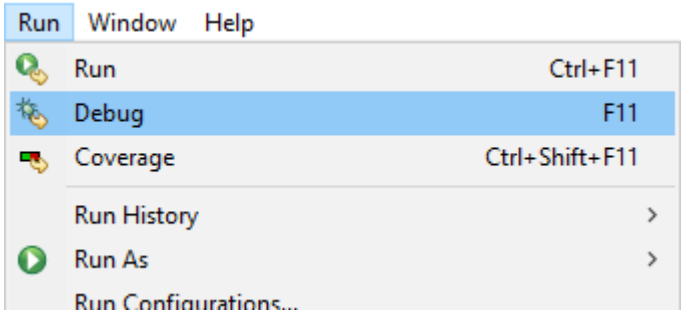
Click the save icon, or Control-S, to save your file.

Congratulations! You have created a HelloWorld project.

Step 4: Running the Project

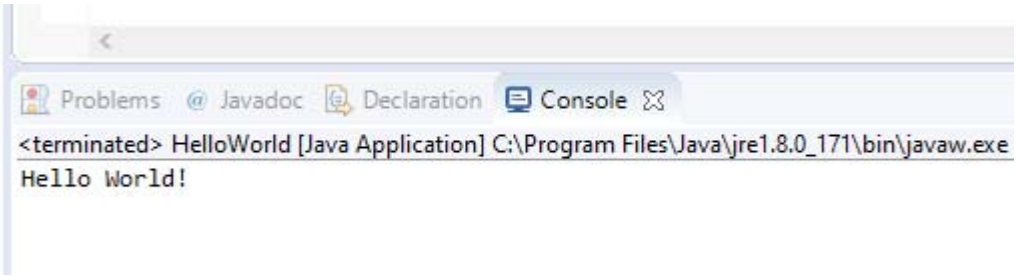
Debug the Project

Once you have your class created, you can choose to “Run” it or “Debug” it. While developing a class, it is recommended to debug it, because you can use breakpoints and other debugging features, and Eclipse will help you figure out where exceptions are occurring rather than just printing them in the output window. Click Run/Debug in the menu to start.



View Output

Your HelloWorld class will execute, and you see this in the output.



Congratulations! You have executed your first class and have seen its output.

Exporting a Project for Submission

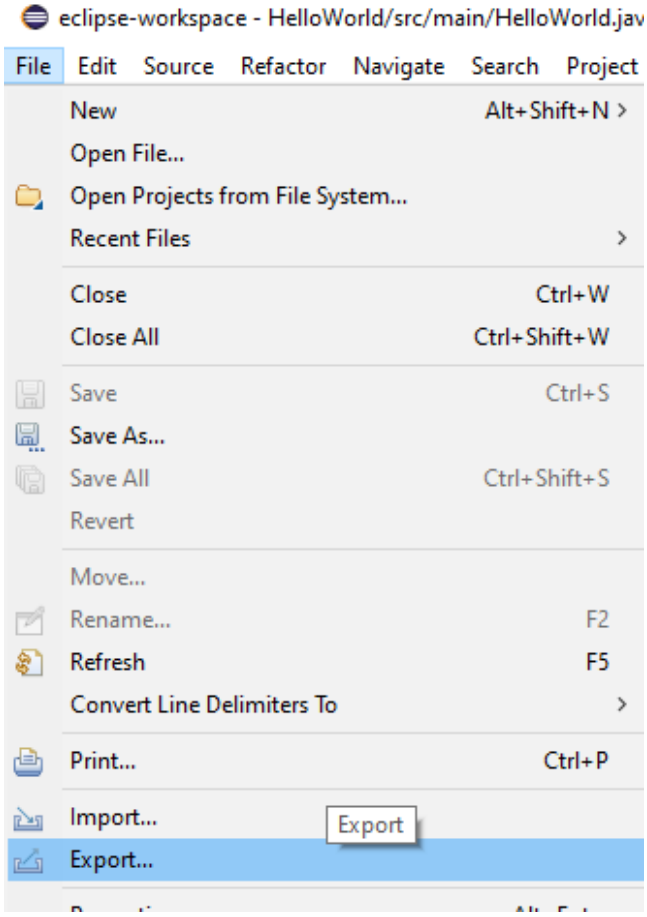
Step 1: Exporting the Project

Understand Exporting

In order to submit your project to your facilitator, you will need to export it, which means you gather its contents into a single location, useful for giving it to others. While there are many formats you can use for export, for classes at BU, you should export into a zip file, because it is a universal file format that virtually anyone can use. Exporting is a better option than manually navigating to your workspace and zipping it manually, because Eclipse performs additional steps to ensure that your export is universally readable by anyone with a similar Eclipse version.

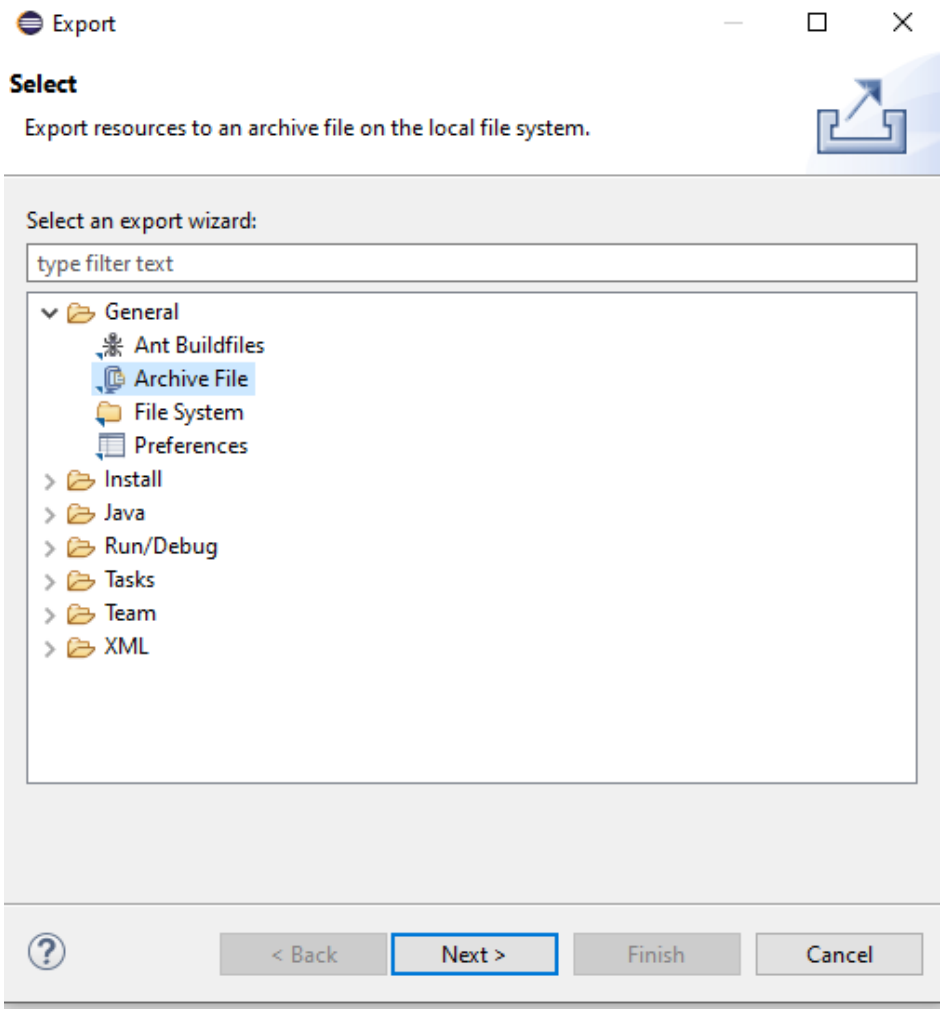
Start the Export

To start the export, click the File/Export… option.



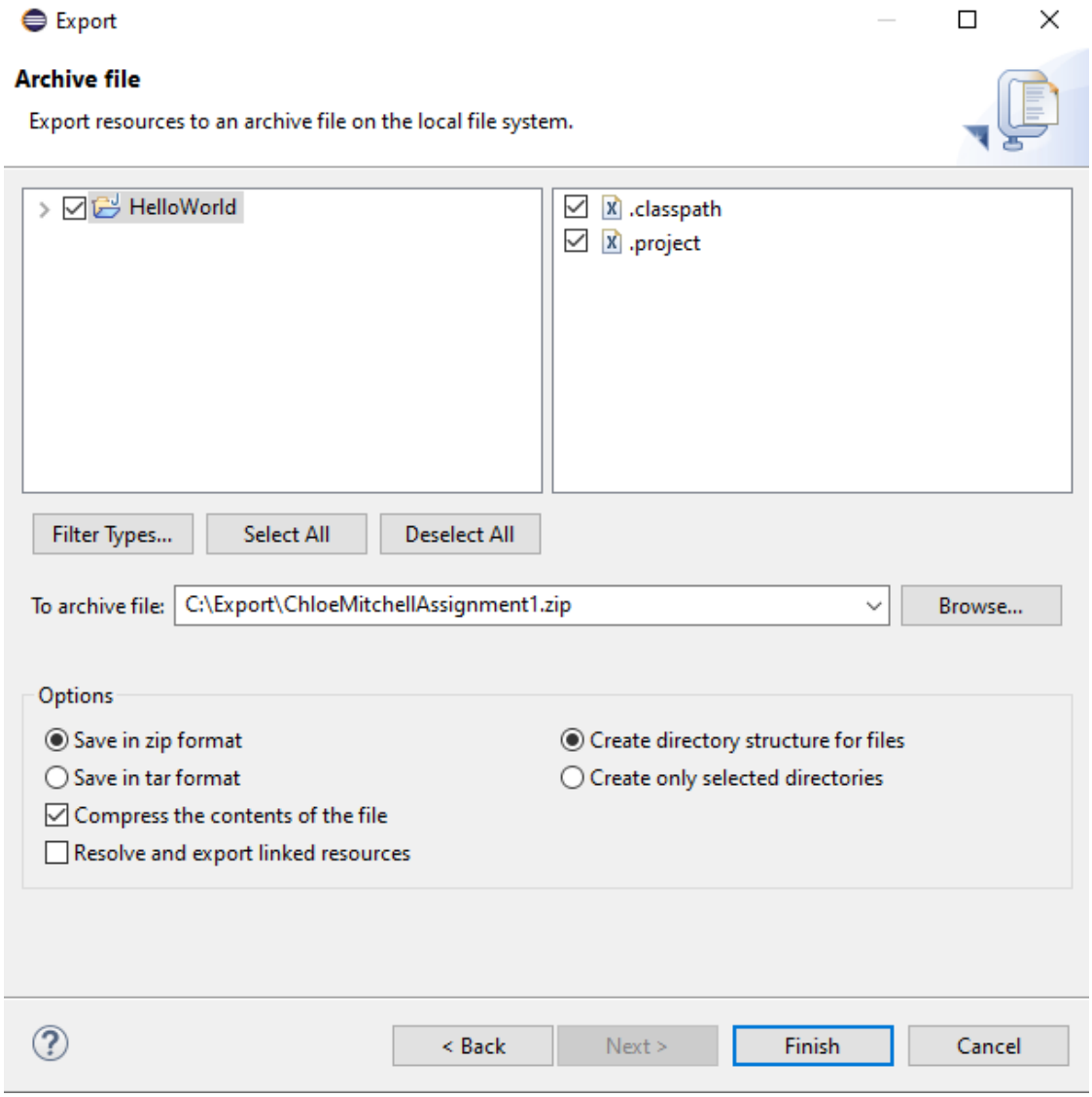
Choose Zip

To ensure the export is a zip file, choose General/Archive File in the list of options that appears on the next screen.



Choose Export Options

After clicking the “Next” button, a screen appears where you set the options for your export.



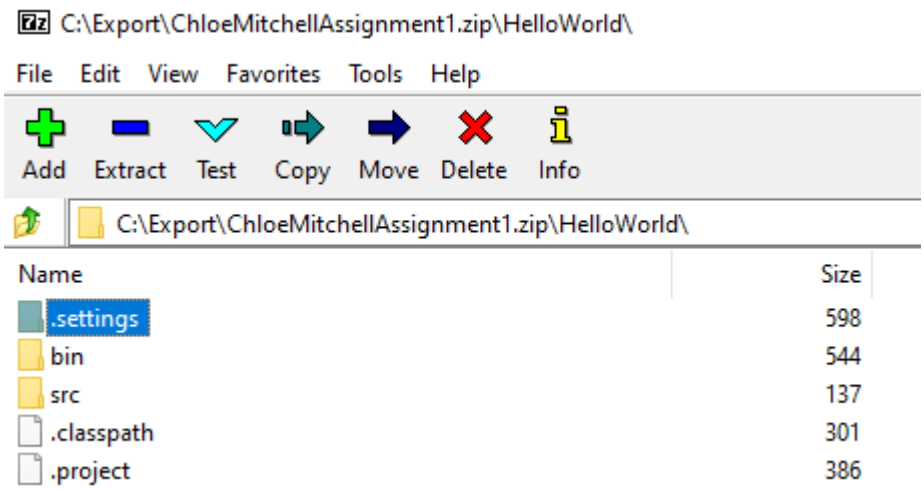
Check the “HelloWorld” checkbox in the first pane, to select all contents in the project. You can export only sub-portions of a project, but that is not recommended. Just export everything in your project every time so everything is included.

You will also be asked to choose the file location in the “To archive file” option. Browse to a directory you can remember, and enter a good filename. A good filename includes your first and last name and the name of the assignment. In this example, we used “ChloeMitchellAssignment1.zip” as the filename, as a submission from Chloe Mitchell for Assignment 1.

Optional Step 2: Viewing the File

View the File

After clicking the “Finish” button to create your export archive, you can optionally view the contents of your zip file. Once you are comfortable with this process, you can skip this step. In our example, the contents of the HelloWorld export look like this.



The .classpath and .project files are needed by Eclipse to understand the contents of the project and what classpath settings to use. The .settings directory contains additional settings for Eclipse. The bin folder contains any class and jar files created in the project. The src folder contains the source files needed by the project.

As you become more advanced with Eclipse, you may rename these directories and add additional directories. For a simple application such as HelloWorld, the above is what you will see.

Congratulations! You can now submit your export as part of your assignment in your class.

Next Steps

You have now downloaded and installed Eclipse, created and executed a simple HelloWorld project, and exported it for submission. This is a great first step! As you start your class, your instructor will ask you to create additional projects. Don't worry. If you can create one project, you can create many.

In the HelloWorld example shown in previous sections, we intentionally kept things very simple. As you work on larger projects, you may need to use other features of Eclipse. You may also need to add additional libraries. Because Eclipse is so popular, there is a plethora of help documentation, and even step-by-step tutorials, on the web for almost anything you want to accomplish. There are also good books on the subject if you want to dive deep. And of course, your facilitator and instructor are available to help you should you get stuck. Take advantage of all of this help to succeed!