How to download, Install and Test JDK, JRE, and Eclipse

If you have installed JDK with JavaFX support and Eclipse using Java in your computer, you don't need to install anything and you are ready to go in this course. Otherwise, please follow the steps below to download, install and test your JDK10.0.2 and Eclipse Photon as required to be able to study in this course.
Note: Mac students may have a little different screens and please make adjustment accordingly.

1. Download and install JDK-**10.0.2** from:
   https://www.oracle.com/java/technologies/java-archive-javase10-downloads.html

   Scroll down, you will see a table consisting of downloading options for different platforms and select one you use. **You will download Java SE Development Kit 10.0.2**. This downloading will automatically include JRE. For installation, simply follow the instruction and most time you just click on next to complete the installation. Your JDK should be installed in the default folder C:/Program files/Java with JRE in a separate folder there.
   Note: Please use default options and simply click on Next all way down in the installation; you don't need to change or custom installing.

   It is your responsibility to make sure JavaFX is supported in your Eclipse if you would like to install newer versions of JDK and Eclipse since they don't include JRE.

2. Download and install Eclipse Photon from:

   https://www.eclipse.org/downloads/packages/release/photon/r

   Note: *It is OK to have multiple Eclipse installed in your computer for purpose of study*

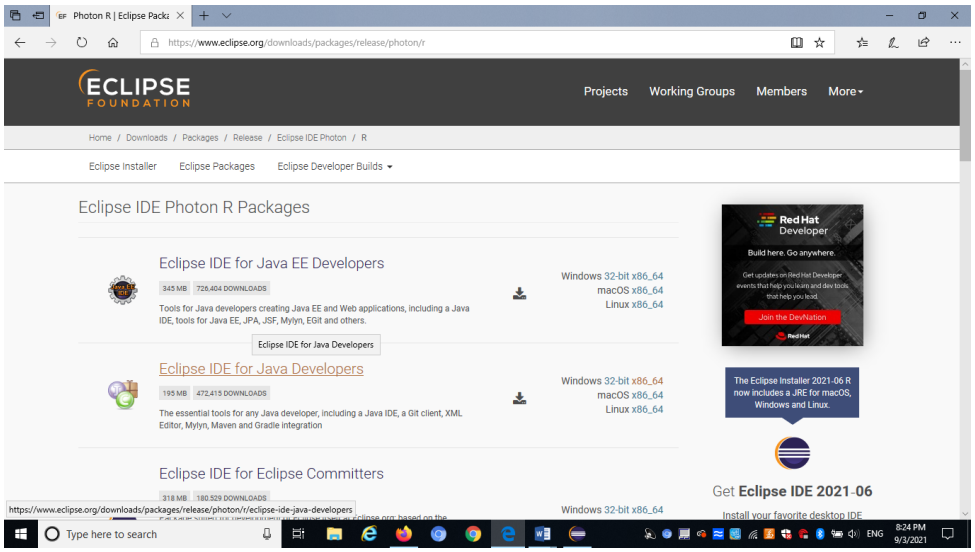

Figure 1. Click on Eclipse IDE for Java Developers to download for Eclipse Photon based on your OS platform (for today's computer mostly Windows x86_64 or MacOS x86_64)

This will let you download and install Eclipse Photon version June 27, 2018 so that you don't need to manually configuring Eclipse to support JavaFX you will learn in the course.

After downloading the file containing the Eclipse IDE for Java Developers in the **eclipse installer**, all you need to do is to follow the instruction to install. Eclipse will automatically create a directory and start shortcut, and save all unzipped files, including eclipse.exe, in this directory. Figure 1 shows that Eclipse IDE for Java Developer is installing (**Screenshots may be vary depending on the version of the Eclipse. Contact instructor for detailed description if you cannot access the figures**).

Note: Eclipse may ask you which JRE you would like to use, you must select JRE10.0.2 you have installed with JDK 10.0.2 in the previous step so your Eclipse will support JavaFX.

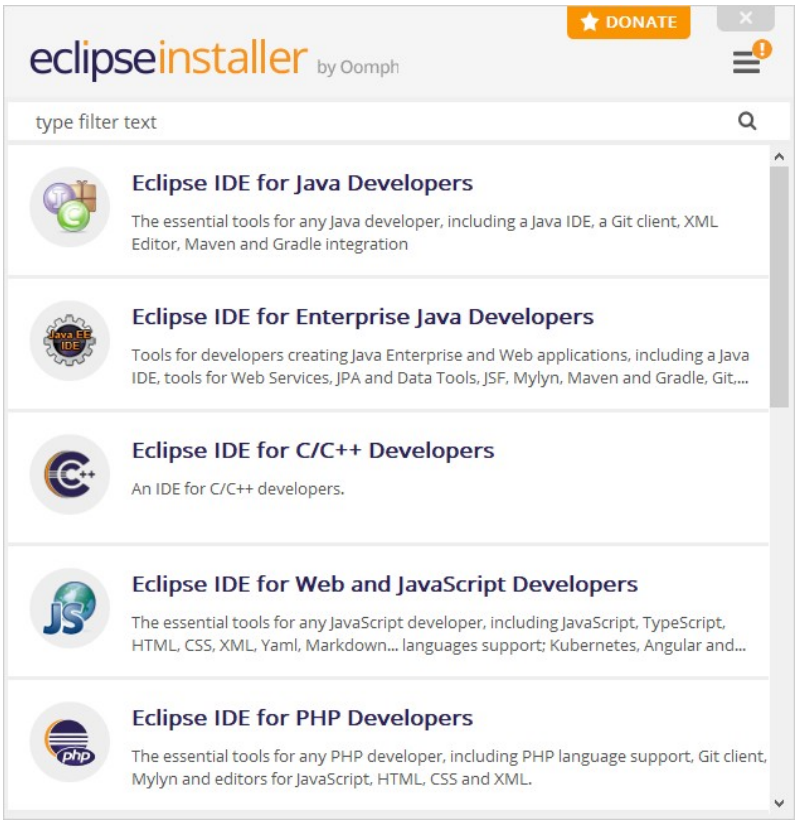

Figure 2. Installing Eclipse IDE for Java Developer. If installer asked, make sure you let installer select JRE10.02 as current VM to support JavaFX in the next step

The following steps must be executed the first time you run Eclipse:

1. Execute Eclipse.exe and make the shortcut on your desktop.

2. In a window that asks you to **Select a directory as workspace**, as shown in Figure 2, type a directory you want to save all labs in this class, say C:\CS125Labs and click on **Launch**.

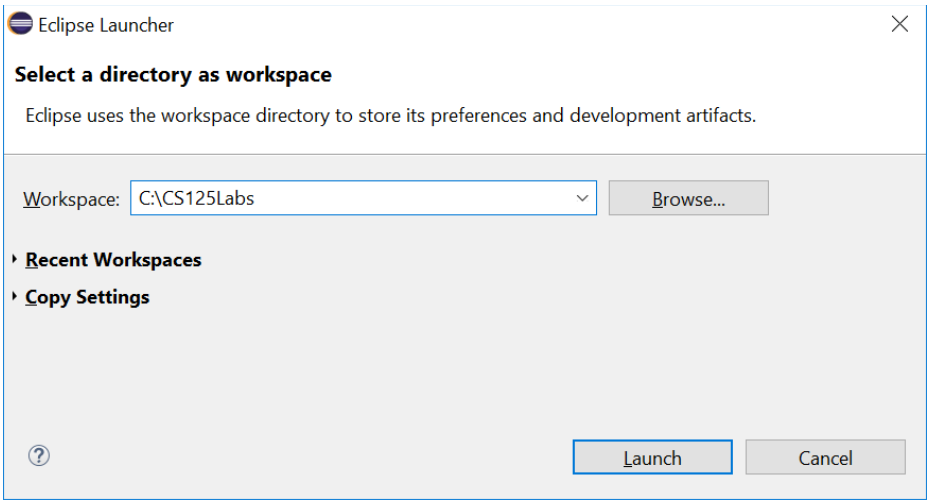

Figure 3. Launch a workspace

3. Eclipse displays a welcome window when a workspace is opened. You may just simply close the welcome window or dismiss it for startup display shown in Figure 4.

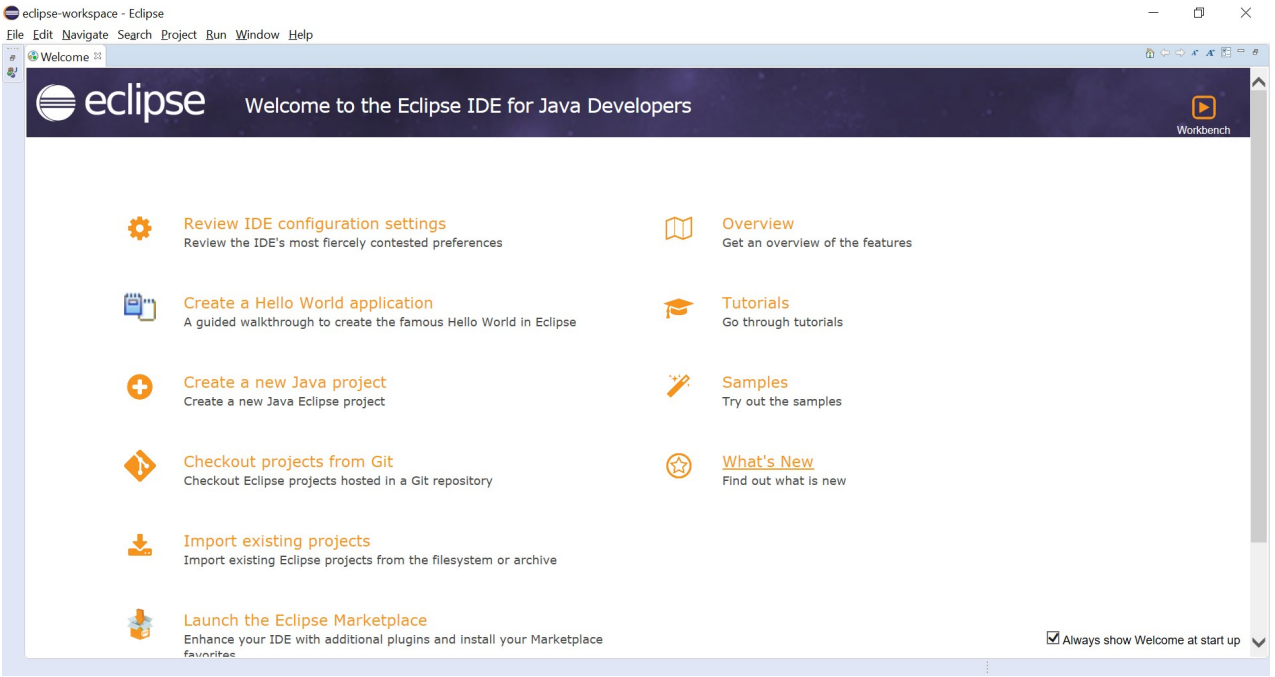

Figure 4. A welcome window in Eclipse

4. Figure 5 shows the first time you open a workspace. Because all files are organized by projects, you must also create a Java project after establishing a workspace. To create a project called Ch1 shown in Figure 5, you may click on **Files** → **New** → **Java Project**. Type your project name, say Ch1, then click on **Finish** shown in Figure 5. A java project named Ch1 is created. (You may also click on the triangle beside the first icon under the File menu to so exactly the same in creating project and java class).

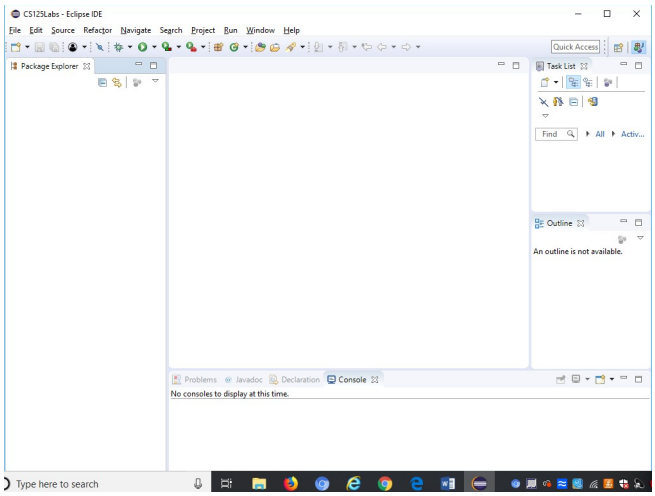

Figure 5. First time open a workspace

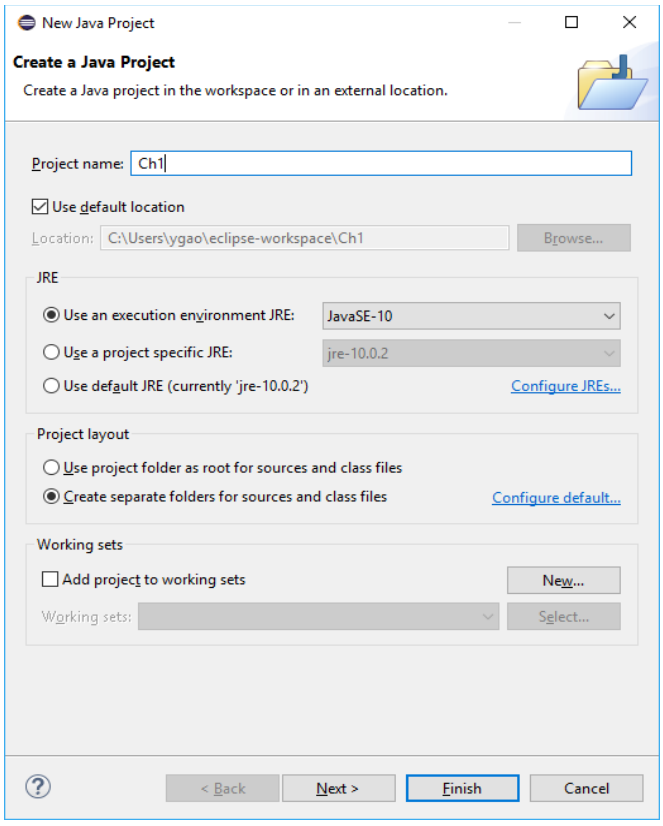

Figure 6. Create a Java project

5. Click on **Don't Create** in the window "Create module-info.java" shown in Figure 7. We will learn how to use module-info later.

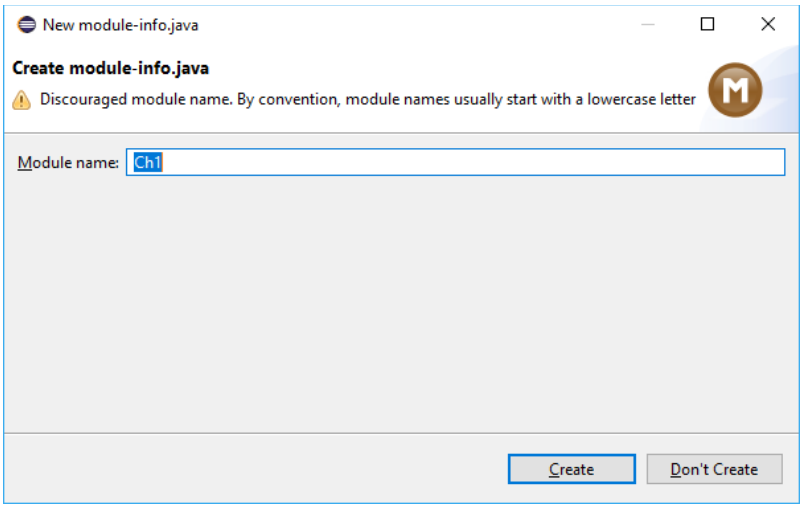

Figure 7. Don't Create module-info.java

6. The next level of file management in Eclipse is to establish a package. For practice purpose, you may just use default package for now and you will learn packages later (omit the warning for now). Click on project Ch1, then click **File** → **New** → **Class**, in the **Name:** field, type your class name, say HelloApp as show in Figure 6. Since HelloApp is a driver class for demo purpose, so check **public static void main(String args[])** as also shown in Figure 8.

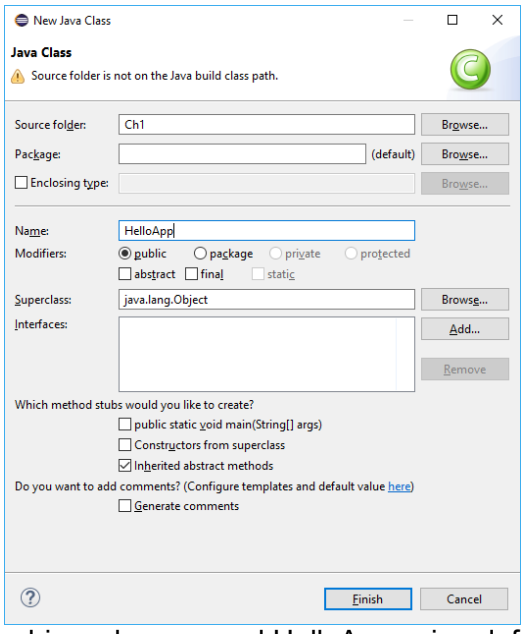

Figure 8. Create a driver class named HelloApp using default package

Now you can enter your code in the editing window as shown in Figure 9.

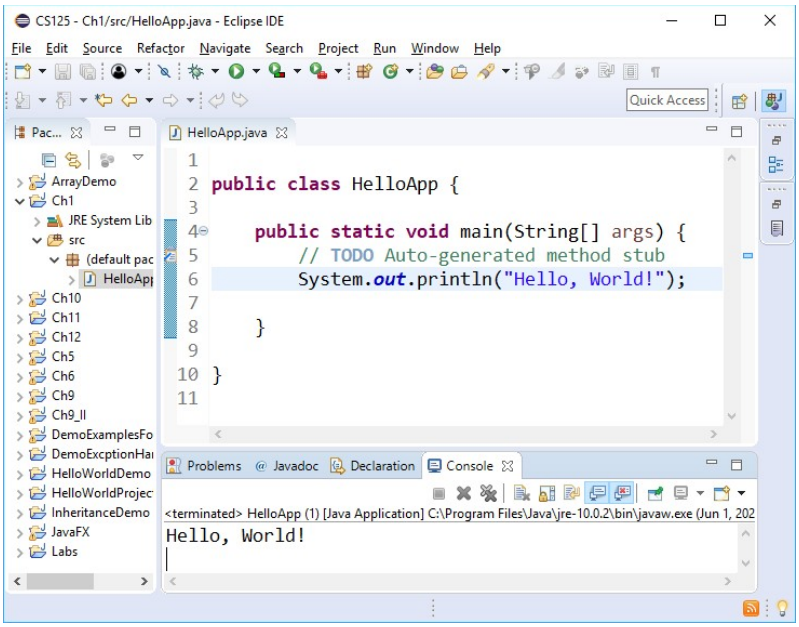

Figure 9. Enter your code in the editing window

To run this demo code, simply click on **Run** menu or run icon as shown in Figure 9. The execution result is displayed in **Console** window at bottom.