Installing Java and Eclipse

0. General.

   a. The instructions below deal mostly with installing software on Windows. If you are installing it on OS X (Apple), Unix, Linux, or some other operating system you will need to make appropriate changes.

   b. I install most Java software in a separate folder under `C:\Java`; e.g., I install the JDK in `C:\Java\jdk-14`, I install JavaFX in `C:\Java\javafx-sdk-14`, I install Eclipse in `C:\Java\eclipse\java-2020-03`, etc.

   c. Several of the steps below involve creating or editing environment variables. In all cases I recommend creating/editing system environment variables, not user environment variables. Also, use caution when editing environment variables, especially `PATH` and `CLASSPATH`. Remember that ';' is the separator for entries in Windows, but ':' is the separator for entries in Linux/Unix. (See separate handout on editing environment variables.)

   d. You should be running a 64-bit version of Windows, and there are 64-bit versions of Java, JavaFX, Eclipse, etc. Sometimes you will need to search for the 64-bit version, and sometimes it will be installed by default. In general, you should run only 64-bit Java.

1. Java Development Kit (JDK)

   a. If you have an outdated version of Java installed, remove it. For versions of Java 10 and lower, you will likely need to use Control Panel to uninstall it. When using Control Panel, you should uninstall all versions of the JDK and the JRE.

   b. Download the latest version of the Java Development Kit (e.g., JDK 14) from http://jdk.java.net/. The file name will be something like `openjdk-14_windows-x64_bin.zip`.

   c. Unzip the JDK file to a directory on your computer . I recommend that you not put it in any directory with spaces in the name. Instead, create a separate directory `C:\Java` and unzip it there. After unzipping, the directory containing the JDK should be something like `C:\Java\jdk-14`. Check this directory to make sure that it contains several subdirectories including subdirectories named `bin` (`C:\Java\jdk-14\bin`) and `lib` (`C:\Java\jdk-14\lib`).

   d. Optional. For future reference, save links in your browser to both
   – the API documentation
   (https://docs.oracle.com/en/java/javase/14/docs/api/index.html)
   – the Tool Specifications
   (https://docs.oracle.com/en/java/javase/14/docs/specs/man/index.html)

   e. Create an environment variable named `JAVA_HOME` with the value of the directory where the JDK is installed; e.g., `C:\Java\jdk-14`.

f. Add `%JAVA_HOME%\bin` to the PATH environment variable.

g. Test your installation by typing "`javac -version`" at the command line. The response should be something like "`javac 14`". Also, type "`java -version`" at the command line, and you should see a similar but more detailed response.

2. Eclipse

a. Make sure that you have already installed Java JDK (see item 1 above) first.

b. Currently, the latest release for Eclipse is 2020-03. Download Eclipse installer from https://www.eclipse.org/downloads/packages/installer – the file will be named something similar to `eclipse-inst-win64.exe`. Double clicking this file will present you with several install options. If you don't know which one to select, I recommend starting with Eclipse IDE for Java Developers.

c. I recommend that you configure the workspace so that source files are kept in one folder named "`src`", and class files are kept in a different folder named "`classes`". To do this, go to Window → Preferences → Java → Build Path. Select the option for Folders, and then name the source folder "`src`" and the output folder "`classes`". (For some reason Eclipse wants to name the output folder "`bin`" by default, but this naming convention is different from that of most other applications that create or use class folders.)

d. Feel free to configure other parts of Eclipse as well. For example, you might want to configure Window → Preferences → Java → Code Style → Formatter to format your source code using a style other than the default. Once you have configured the formatter, you can export the settings to an XML file and then later import them on a different installation of Eclipse.

e. I recommend placing a shortcut to `C:\Java\eclipse\java-2020-03\eclipse\eclipse.exe` somewhere on the desktop.

f. Test your installation by double clicking on the shortcut or directly on the file `C:\Java\eclipse\java-2020-03\eclipse\eclipse.exe`.

3. JavaFX (Optional)

a. Download the latest version (currently 14) of JavaFX from https://openjfx.io/. The file name will be something like `openjfx-14_windows-x64_bin-sdk.zip`.

b. Unzip the file to `C:\Java`. Double check that `C:\Java` contains a subdirectory named similar to `javafx-sdk-14` (`C:\Java\javafx-sdk-14`), and that this subdirectory has several subdirectories, including one named `lib` (`C:\Java\javafx-sdk-14\lib`).

c. Create an environment variable named `JFX_HOME` with the value of the directory where JavaFX is installed; e.g., `C:\Java\javafx-sdk-14`.

d. Create a User Library in Eclipse: Window → Preferences → Java → Build Path → User Libraries → New. Name it JavaFX 14 and include the jars under the `lib` subdirectory for JavaFX 14 (`C:\Java\javafx-sdk-14\lib`).

e. Add JavaFX 14 to any Eclipse Java project that needs to use JavaFX. Right mouse click on the project name and select Build Path → Configure Build Path… → Libraries (tab) → Modulepath → Add Library… → User Library → Next → select JavaFX 14 → Finish

f. When running an application from Eclipse, you might see the error message:
Error: JavaFX runtime components are missing, and are required to run this application
If you encounter this message, the take the following actions starting with menu item Run: Run → Run Configurations… → Arguments (tab). In the text box labeled "VM Arguments:" enter
"`--module-path=${env_var:JFX_HOME}\lib --add-modules=javafx.controls`"