Installing Eclipse + CDT

Step 0:

If you have not installed or have removed Java's JRE and JDK from your computer, you will need to reinstall java. This course is taught in C++. However, the IDE (Eclipse) is

Windows:

Step 1: You must install a c++ compiler onto your PC.

You have two choices: **MinGW GCC** or **Cygwin** with gcc. If you are already familiar with unix and want to have a unix environment on your PC, feel free to install Cygwin (I personally love and miss unix). Make sure when selecting what to install, you install gcc and related files.

However, most you may prefer to Install MinGW GCC because it is simpler and more straightforward.

To install MinGW:

1. Go to MinGW site at https://sourceforge.net/projects/mingw/files/ . Choose downloads, then Installer, then mingw-get-setup.exe.
2. Run the download installer.
   Set the installation directory. You can use the default directory of C:\MINGW. Don't use either desktop or Program Files (or any folder with a space in the name.
3. In MinGW Installation Manager, make sure Basic Setup is chosen. If, to the right of Basic Setup, everything isn't selected, make sure you select everything except ADA, Fortran, and CSharp. Then Select Installation->Apply Changes. This step might take a while.
4. Set your environment variable PATH to include "`<MINGW_HOME>/bin`" where `<MINGW_HOME>` is the MinGW installed directory that you have chosen in the previous step. (probably C:\MinGW). You may have to google how to set environment variables on your computer. On windows 10, go to settings then type in control panel. Select system->advanced system settings->advanced->environment variables. Under System Variable, Choose the Path variable, and then choose edit. Add to the end, C:\MinGW\bin (no slash or ; at the end unless you're adding something else to the path as well). Then hit a lot of ok's.

Step 2: **Install Eclipse + CDT**

This should be fairly straightforward.

1. Go to https://eclipse.org/cdt/, click downloads, scroll down to Eclipse IDE for c/c++ developers, and choose the version appropriate for your computer to download (downloading may take a while).
2. Move the zipped file to a folder (perhaps C:\classes\220.
3. Unzip the file.
4. Launch Eclipse.

Note: if you are unable to get Eclipse to work with MinGW, you can try installing Code::Blocks. I have not installed this, but this only requires that you download and install Code::Blocks without first installing a separate gcc compiler. This can be found at: http://www.codeblocks.org/

Mac:

You may be able to skip Step 1, because many Mac machines come with a c++ compiler (MAC OSX gcc). So first follow step 2, above, and then try to compile and run your first c++ project (see the next page). If you are unable to then get C++ to compile code within the Eclipse environment, then you may need to install MinGW (See step 1, above, skipping step 4), or, as an alternative to Eclipse, you can download and install Code::Blocks from http://www.codeblocks.org/

Writing your First C/C++ Program in Eclipse

(largely taken from https://www3.ntu.edu.sg/home/ehchua/programming/howto/EclipseCpp_HowTo.html)

Step 1: Launch Eclipse

1. Start Eclipse by running "`eclipse.exe`" in the Eclipse installed directory (It may help to pin the eclipse icon to your start menu for future use).
2. Choose an appropriate directory for your *workspace* (i.e., where you would like to save your works). Most likely it will be in the 220 folder you created.
3. If the "welcome" screen shows up, close it by clicking the "close" button.

Step 2: Create a new C++ Project

For *each* C++ application, you need to create a *project* to keep all the source codes, object files, executable files, and relevant resources.

To create a new C++ project:

1. Choose "File" menu ⇒ "New" ⇒ Project... ⇒ C/C++ ⇒ C++ project.
2. Choose C++ Manage Build in the pop-up box
3. The "C++ Project" dialog pops up.
    1. In "Project name" field, enter "`FirstProject`".
    2. In "Project Types" box, select "Executable" ⇒ "Empty Project".
    3. **In "Toolchains" box, choose your compiler, e.g., "MinGW GCC" ⇒ Next if you have installed mingw, or MAC OSX GCC if you are on a Mac.**

       If you are a MAC user and no compiler shows up here, that means you'll have to install MinGW. Get out of this, go back to step 1 in the first part, and install MinGW (skipping step 4: setting the environment variable). Then restart your computer and try this part again.

       Note: If you choose the wrong compiler, you code will not compile, and you won't be able to run it.

4. If the "Select Configurations" dialog appears, select both "Debug" and "Release" ⇒ Finish.

Step 3: Write a Hello-world C++ Program

1. In the "Project Explorer" (leftmost panel) ⇒ Right-click on "`FirstProject`" (or use the "File" menu) ⇒ New ⇒ Source File.
2. The "New Source File" dialog pops up.
    1. In "Source file" field, enter "`Hello.cpp`".
    2. Click "Finish".
3. The source file "`Hello.cpp`" opens on the editor panel (double-click on "`test.cpp`" to open if necessary). Enter the following codes:

```
#include <iostream>
```

```
using namespace std;

int main() {
   cout << "!!!Hello, world!!!" << endl;
   return 0;
   }
```

Step 4: Compiling and Running your Code

1. At the top, under project, click on Build Project. Build All builds all your projects, not just the one you are currently working on.
2. Make sure you select Local App
   A. You should have no errors. If you have “unresolved” errors, it most likely means you haven’t put the correct path into your system variable. Go back to step 4 in the installation process and retype the environmental path variable. If you are on a Mac, you may need to install MinGW at this point.
   B. If you have an error that’s something along the lines of, “unresolved cl.exe”, either you haven’t set your path properly (see above), or you chose the wrong compiler in the tool chain. Under Project, choose Properties. Under C/C++, choose Tool Chain. Under Current ToolChain choose the appropriate c++ compiler (most likely MinGW gcc or MacOSX gcc). Hit Apply and Ok.
   C. If the error is something else, then most likely it’s a typo. Recheck your code to make sure it’s exactly as above.
3. When it builds, you should now have a binary in the left column. Under Run, choose Run.
   You should get:
   ```
   !!!Hello World!!!
   ```
   In the console window

   Note: if you chose the wrong compiler, you can always go back and change it. Under Project->Properties, select C/C Build->Tool Chain. And then under Current toolchain choose MinGW GCC.

Congratulations! You have just successfully compiled and ran your first c++ program.