

C++ for Eclipse Tutorial by Steve Cole, Spring 2015

The goal of this tutorial is for you to have a fully functional C++ development environment in Eclipse and to set up Lab 0 from your class repository in this environment. We will assume you are working on a Windows 7 machine and have already installed Eclipse with Java support. If you have not yet installed Eclipse, follow the instructions in [this tutorial](#) (click 'Installing Eclipse' from the left menu). If you are using a different operating system, you may have to adapt these instructions slightly; in particular, you will need to install a compiler that is compatible with your system.

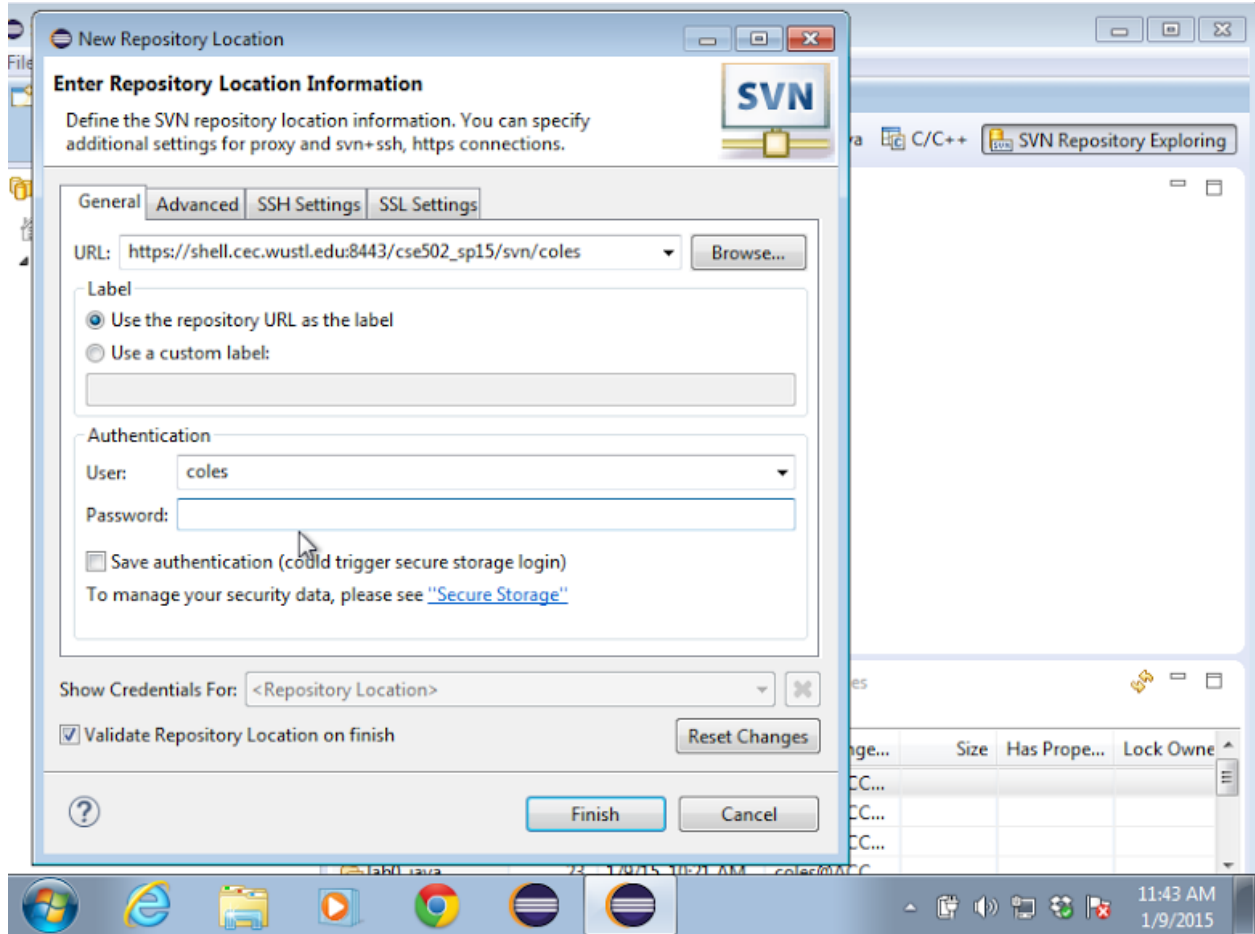
1. The first thing we need is a C++ compiler; several of these exist, but the grading scripts use one called gcc/g++. Follow the instructions [here](#) to install Cygwin, which is a Windows utility that contains gcc/g++. Note that in Step 2 of the instructions, it is sufficient to select gcc, g++, make, and gdb. Step 4 is optional.

2. Now that we have a compiler, we need to add C++ support to Eclipse so that it can work with C++ source code. Follow [these instructions](#) to install the "CDT" plugin for Eclipse (if the URL for acquiring the CDT plugin does not appear when you type "CDT" into the "Work with:" bar, enter this URL: <http://download.eclipse.org/tools/cdt/releases/8.5>).

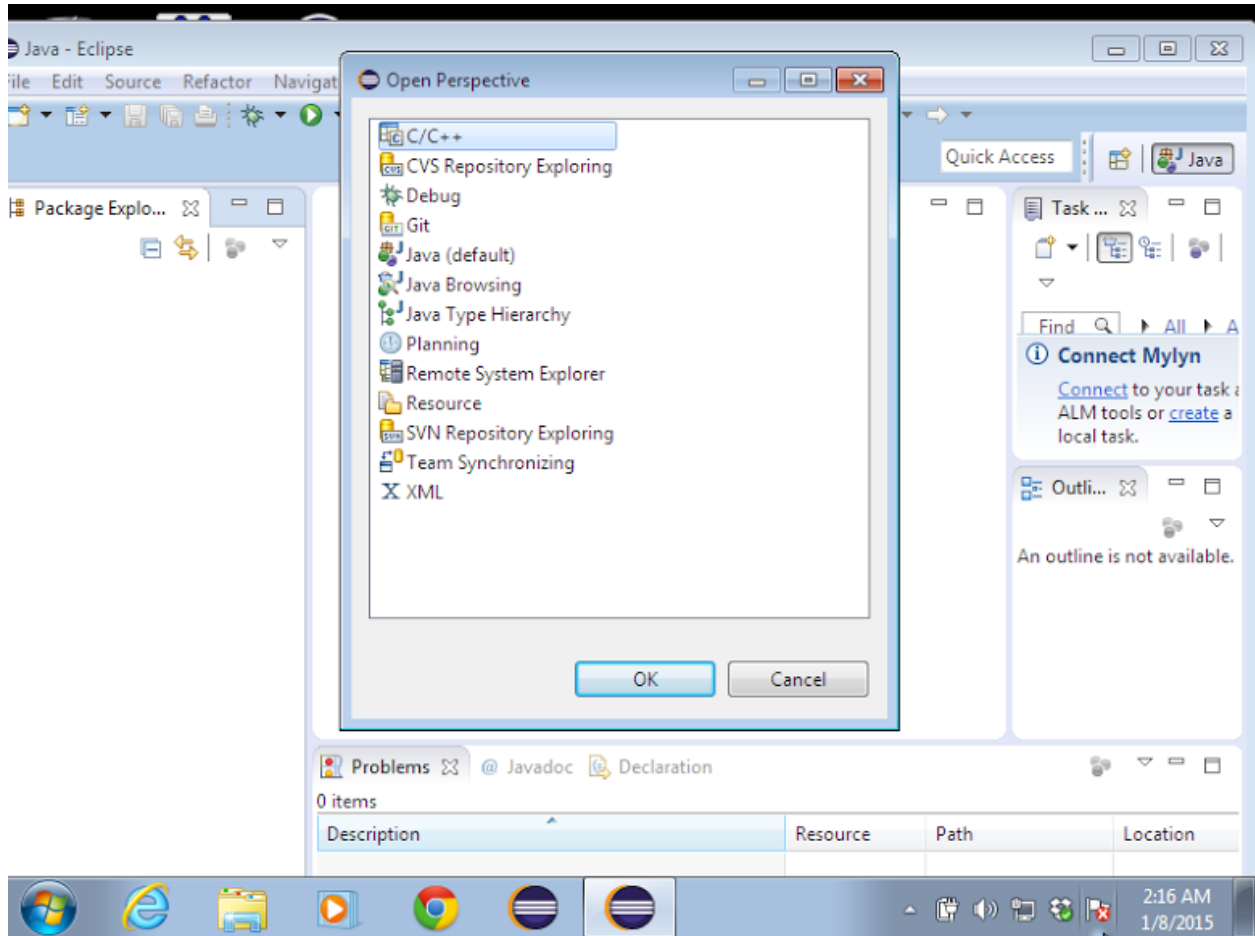
3. Follow the instructions from the [CSE 131 class website](#) [<http://classes.engineering.wustl.edu/cse131/>] to set up Subversive in Eclipse, which is the tool that allows you to access your SVN repository. When you're on the CSE 131 main page, find the two links "Installing Eclipse" and "Using subversive" in the left menu under "Tutorials", and follow the instructions there starting with "Installing Subversive" halfway down the "Installing Eclipse" page. Note that the SVN Version number you see may differ from the one on the Tutorial, but just choose the latest version shown.

Remember to configure the repository for cse502_sp15 instead of cse131_fl14. Your repo path should be:

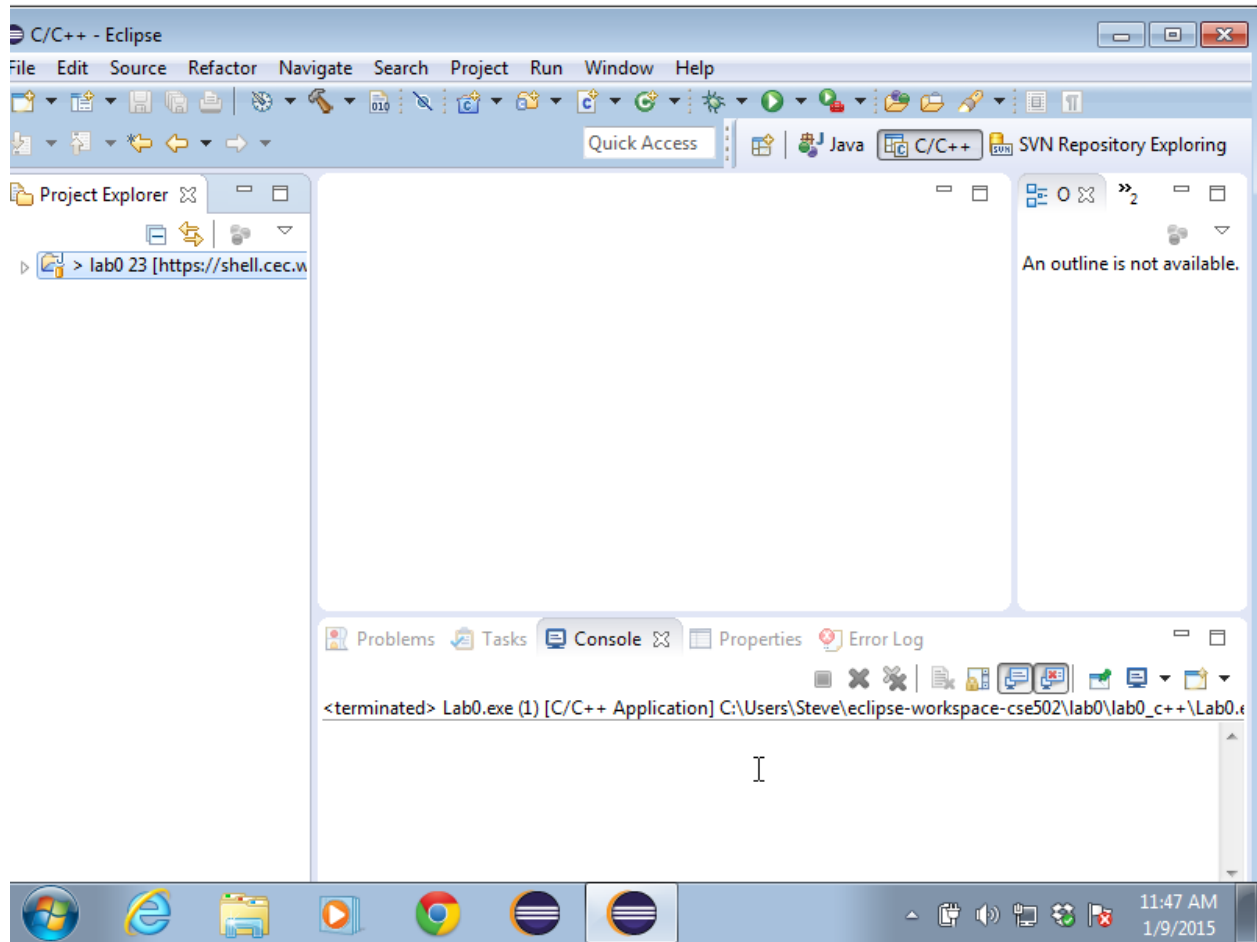
https://shell.cec.wustl.edu:8443/cse502_sp15/svn/<Your WUSTL key> .



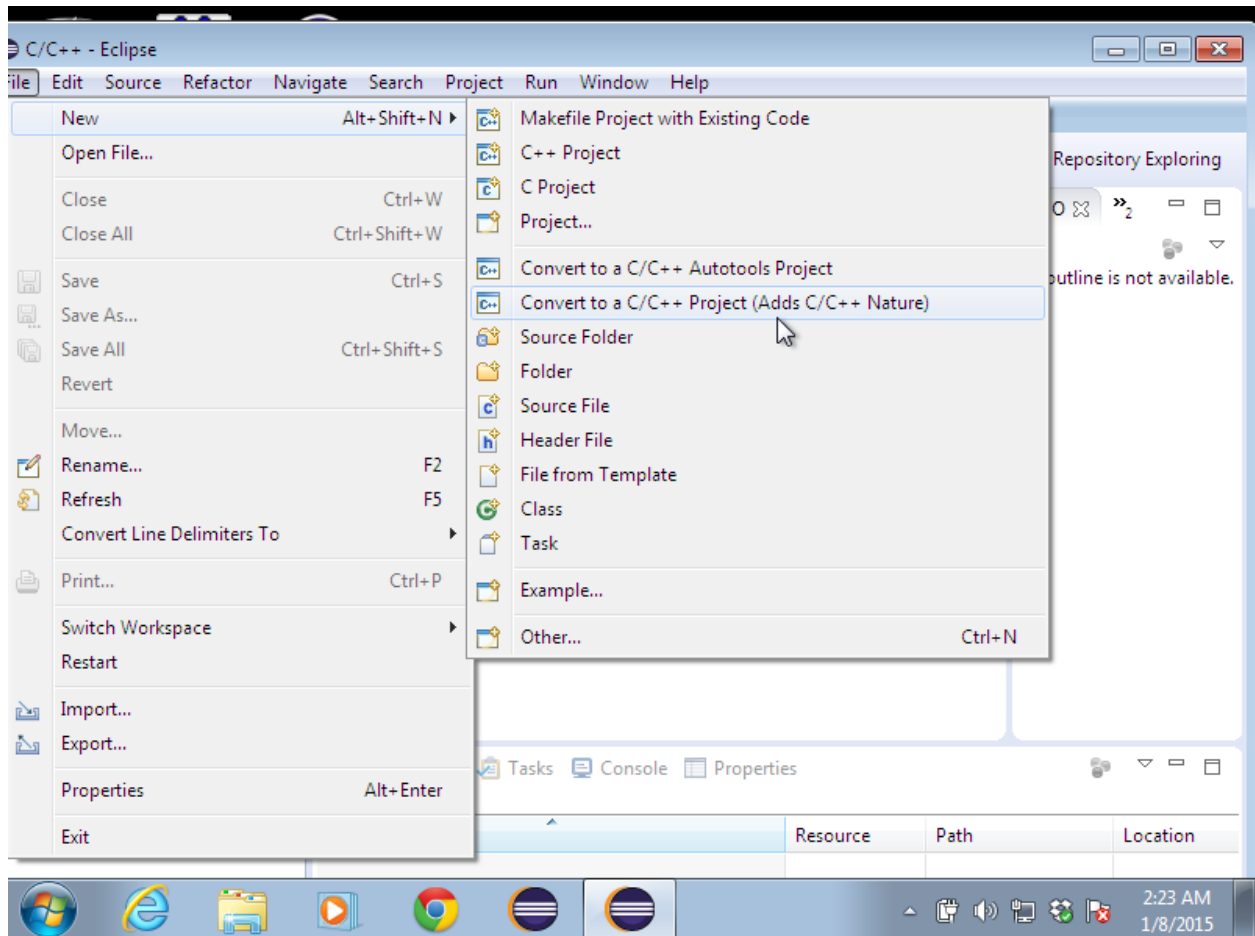
4. Now we're ready to open Eclipse and configure our first lab assignment. Open the Eclipse workbench, then click on "Open Perspective" in the top right of the window, next to the "Java" icon. Open the "C/C++" Perspective, as well as the "SVN Repository Exploring" perspective.



6. Check out the lab0 project from your repository and open it in the C/C++ view by clicking on the C/C++ icon in the top right of the Eclipse window.

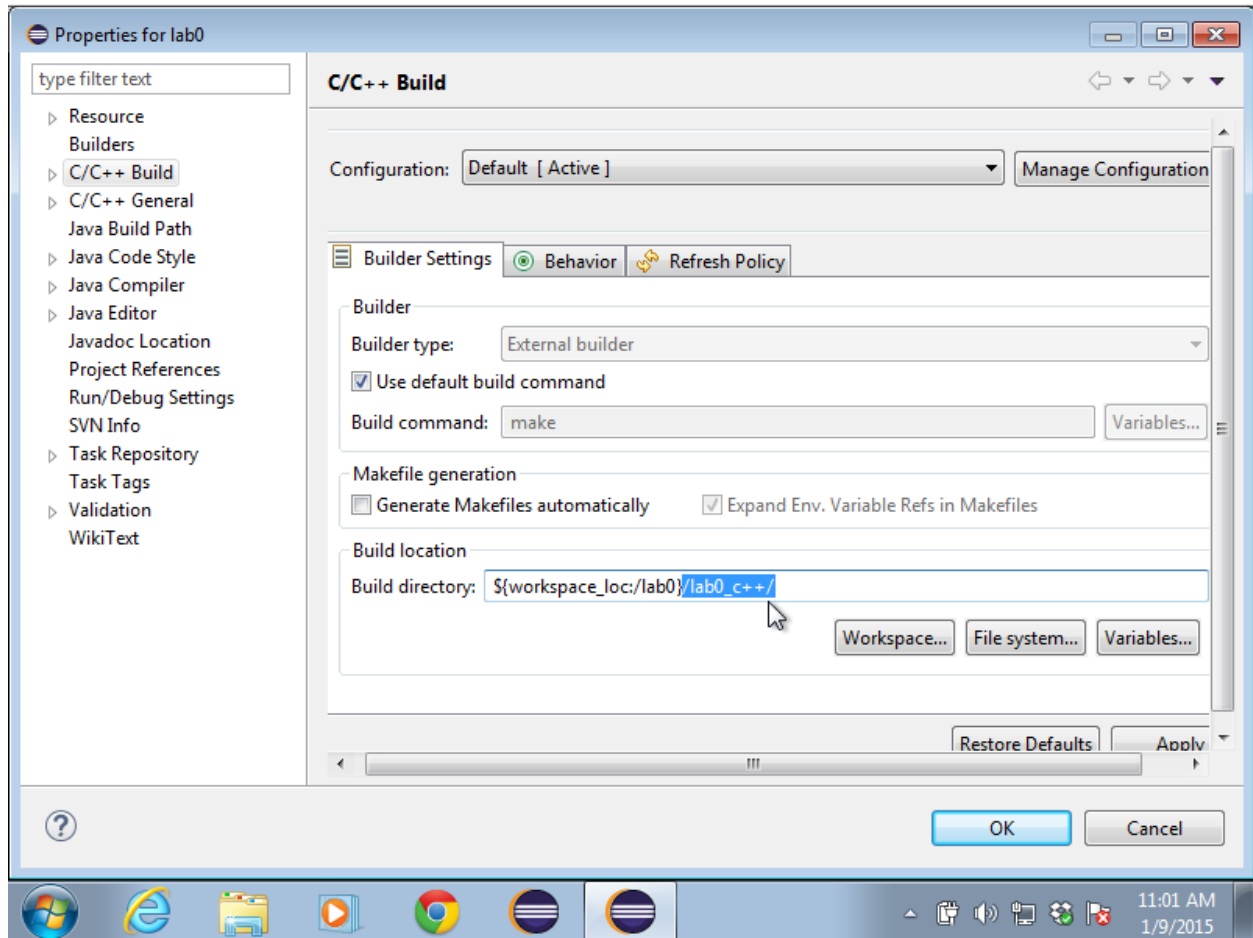


7. With the lab0 project folder highlighted, select **File -> New -> Convert to C++ Project**.



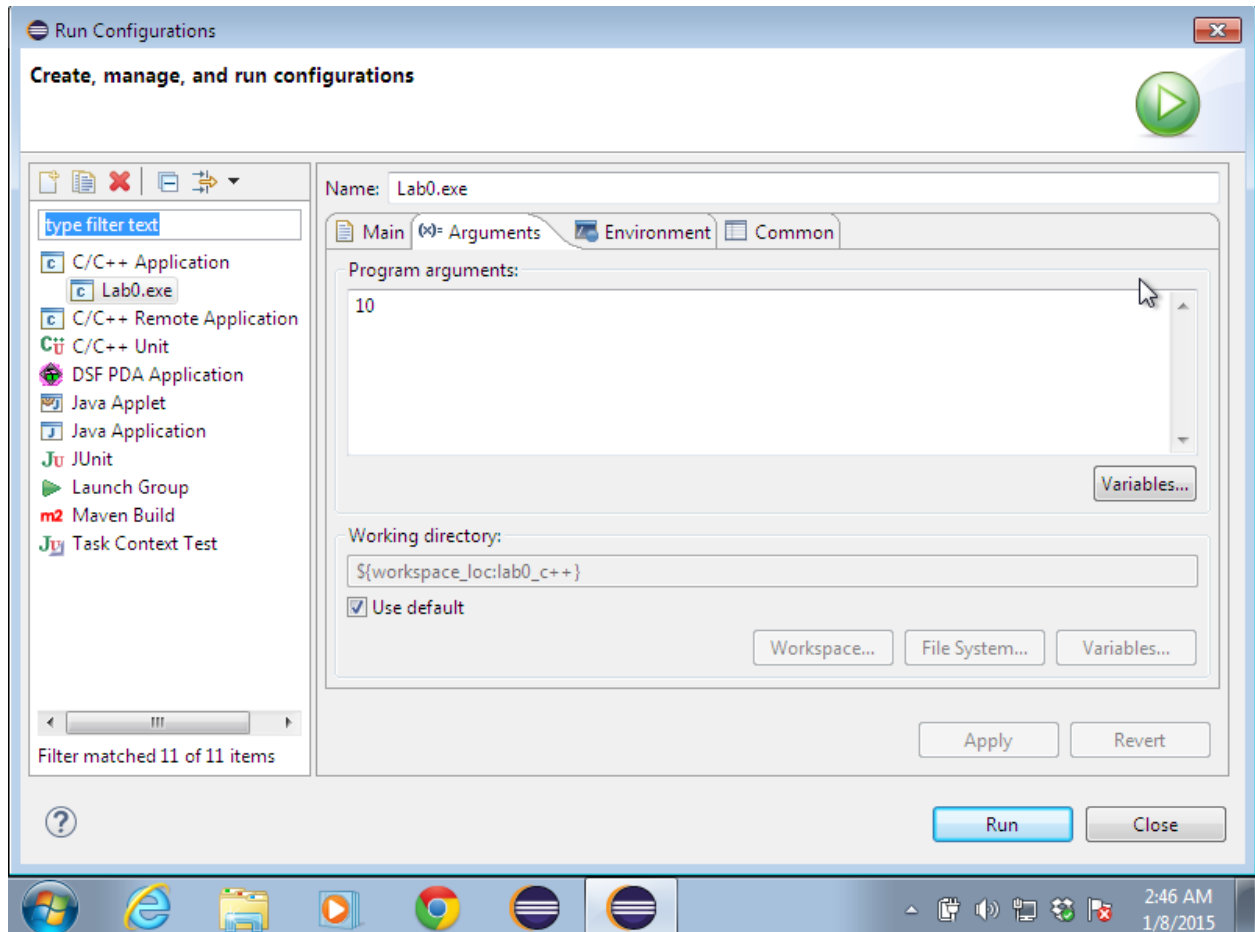
Scroll down and select **Makefile project** as the project type, and select **Cygwin GCC** as the project toolchain.

8. The Lab 0 project contains folders for both Java and C++ implementations, so we need to tell Eclipse where it can find the C++ source code and build instructions. Right-click the lab0 project folder and select **Properties** (at the bottom of the list), then select **C/C++ Build** from the list of properties on the left side of the window. Under the **Build location** heading, append "lab0_c++/" to the **Build directory** setting.

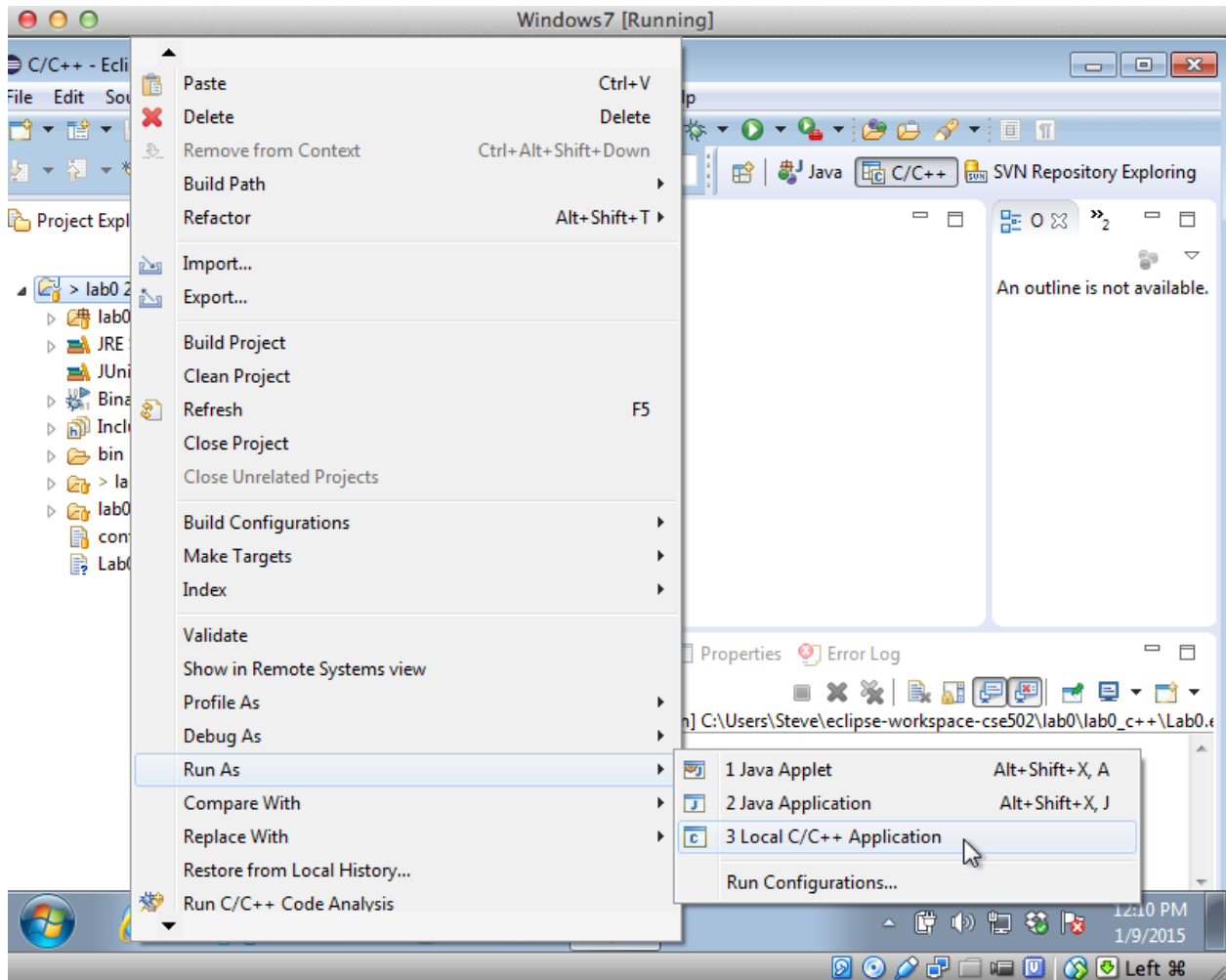


9. The project is now set up for you to start developing, but first let's verify that we can build and run the project framework successfully as-is. To compile the code, right-click the project folder and select **Build project** . (Alternatively, click the hammer icon in the main toolbar.) The project should build successfully with no errors; if it does not, troubleshoot any errors before moving on.

10. If the code compiled successfully, you should see a few new files in the lab0_c++ folder. Now we're ready to set up the runtime options and run the program. Right-click the newly-created **Lab0.exe** file and select **Run as -> Run configurations...** . Select the **Arguments** tab; anything you enter in the text box will now be passed to the program as a runtime argument. In this case, Lab 0 takes a single integer argument.



Click **Run** to run the program. Note that once input arguments have been entered, Eclipse remembers them, and you can run the application by just right-clicking **Lab0.exe** and selecting **Run as -> Local application**. (Alternatively, click the green 'play button' icon in the main toolbar.)



10. If the code ran successfully (no errors), you're ready to start adding your own code to the lab! Follow the instructions in the Subversive tutorial for committing your changes to the SVN repository as you go along. *Remember that checking in changes to the repository is an entirely separate process from saving your work. You should do both often.*