[**Dev-C++ Website**](http://orwelldevcpp.blogspot.com/)

 Installed in ACC labs.

C++11 setting - Go to Tools -> Compiler Options, select Settings tab, then select Code Generation tab, Drop down on Language standard, pick ISO C++11

If get **auto** error, add

-std=c++1y

 to Tools->Compiler-Options->Add the following commands when calling the compiler  (and check the box)

To reset the Dev-C++ profile, go to  Tools->Environment-Options->Directories and Click button to Remove settings and Exit.

[**CLion Website**](https://www.jetbrains.com/clion/)

A cross-platform IDE for C and C++. You will need to install a C++ compiler in addition to the CLion IDE.

You will need to do a free registration with Jetbrains. <https://www.jetbrains.com/student/>

[Install and set up CLion](https://www.jetbrains.com/help/clion/install-and-set-up-clion.html)

CLion requires a project when working with code.  For template classes remove the .cpp file from the CMakeLists.txt file

If get **auto** error make sure that the CMAKE\_CXX\_STANDARD is set to 14 in CMakeLists.txt.

For projects with multiple files that each have a main you will need to modify CMakeLists.txt to have an add\_executable target for each main file. Each target can only have one main file.

**Using Visual Studio 2017**

1. Start Visual Studio 2017
2. File -> New -> Project
3. Choose a template: Visual C++ ->  General -> Empty Project
4. Give a Project Name for the project and pick a folder to save the project in
5. Right Click on Source Files in Solution Explorer to add a new item
6. Select Visual C++ -> C++ File and enter the correct name for the lab assignment
7. Type in C++ code as normal, save work
8. Use the play (F5) button to compile and run the assignment (Or Debug -> Start Debugging on menu)
9. Browse to the project folder to find your assignment file to upload to Blackboard

To use template classes move the implementation in the .cpp file to the include file for each template class.

[Source code organization (C++ Templates)](https://msdn.microsoft.com/en-us/library/mt750459.aspx)

Visual Studio does not support using **auto** keyword as part of a function return in versions earlier than 2017.

Visual Studio does not allow multiple main functions so some of my example code will not work without removing the extra files with main functions.

Do not try to run your project under different versions of Visual Studio, it will break.

[Use of "stdafx.h"](http://www.cplusplus.com/articles/1TUq5Di1/)

**Using Xcode on the Mac**

1. Install Xcode for free from App Store
2. Start Xcode
3. File -> New -> New Project
4. Choose a template: Mac OS X - Application -> Command Line Tool
5. Give a Product Name for the project, Choose C++ for Type, and uncheck Automatic Reference Counting
6. Pick a folder to save the project in
7. Rename the main.cpp file to the assignment file name by double clicking on the main.cpp name
8. Use the run button to compile and run the assignment
9. Use Shift-Command-C to activate the console to interact with the running program and see the results
10. Browse to the project folder to find your assignment file to upload to Blackboard

How can you use auto and/or smart pointers in Xcode?

Build Settings -> Apple LLVM 8.1 Language C++ -> C++ Language Dialect -> choose C++14

How can you get rid of errors with templates and getting multiple definitions?

Build Phases -> Compile Sources -> Only select the file with the main function. Unselect all other files.

Note: Xcode automatically tries to compile all cpp files added to a project. That does not work for templates.