Troubleshooting Common Issues with VS 2017

Inclusions not Recognized

When opening Sample2017 up for the first time, you may be confronted by code that has red underlines on inclusions. If you try to build at this point, a whole bunch of inclusion errors will be reported:

If this is the case, you will want to go to >Project>Sample Preferences in the toolbar:
This should cause a window to pop up. In order to fix your issue, you will need to change the Windows SDK Version to inherit from parent or project defaults.
Be sure to hit APPLY after changing the SDK version:
Once this is done, change the SDK Version back again, hitting APPLY:
When you exit out of the window, the include lines should no longer have red lines under them, meaning that Visual Studio took the hint and now recognizes the inclusions as valid.
Trouble including heli.550 in your project

If you have not already, it is suggested that you get the contents of heli.550 off of the class website:

You then want to add another file to the sources folder:
Unfortunately our “heli.cpp” file is not enough. Visual Studio does not generally like the practice of including source files, preferring the much more idiomatic use of header files. So, we end up with a lot of extra inclusions under the hood due to Visual Studio’s linking scheme and the build fails:

You will need to break your source file into a source file and a header file, for the sake of this example, we’ll just name it “heli.h”:
First, you should move the struct declarations out of the source file and into the header file:

Also, be sure to include the header file in the source file:
Lastly, you will need to declare the variables in the source file in the header file. Be sure to give them the “extern” descriptor:

Once this is done, verify that you are including the header file in both source files and no source files in any of the files, then try to build again. So long as there are no unrelated errors in the project, it should build successfully: