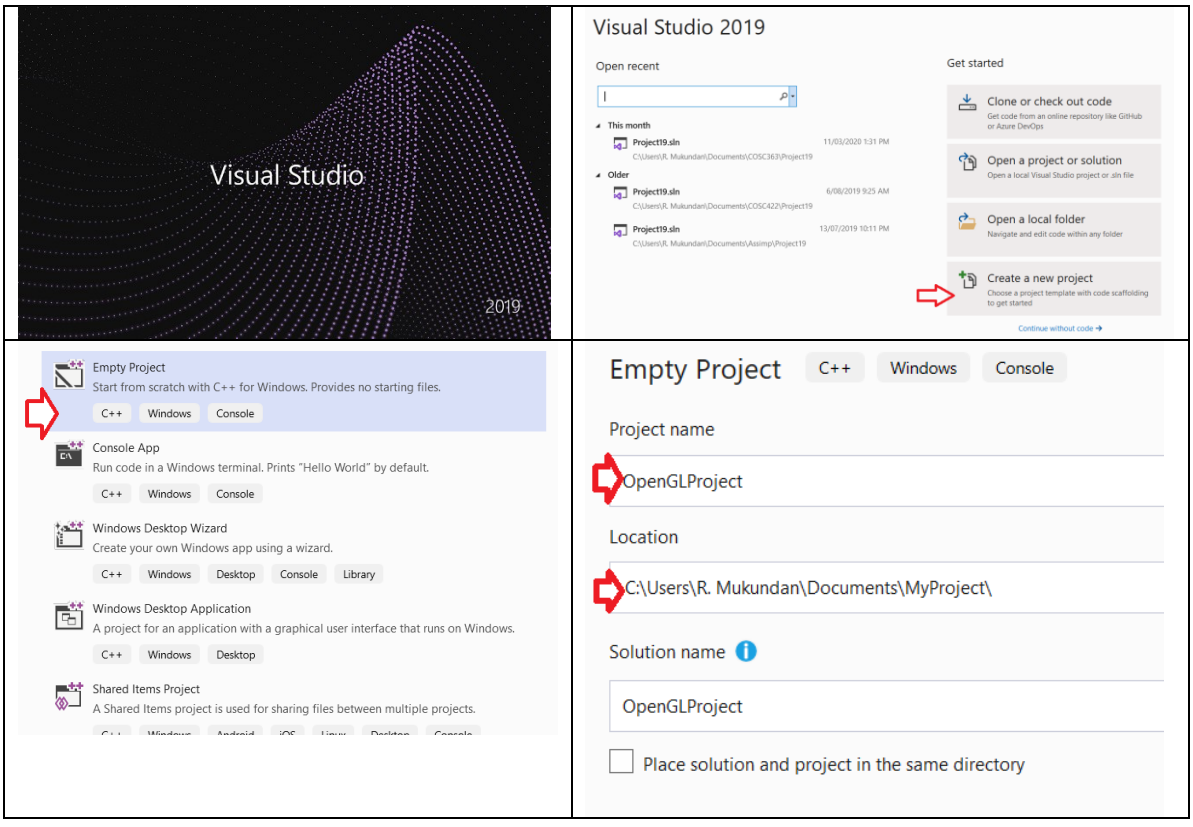
OpenGL Setup for Visual Studio on Windows

R. Mukundan,
Department of Computer Science and Software Engineering, University of Canterbury.

The procedure outlined below can be implemented for different versions of Visual Studio (VS10-VS19) as it does not require files to be stored in default VS directories or system folders. It therefore does not require administrative privileges for copying files to these folders. Even though this document contains 6 pages, it mostly contains screenshots giving a step-by-step explanation, and the procedure is not complex!

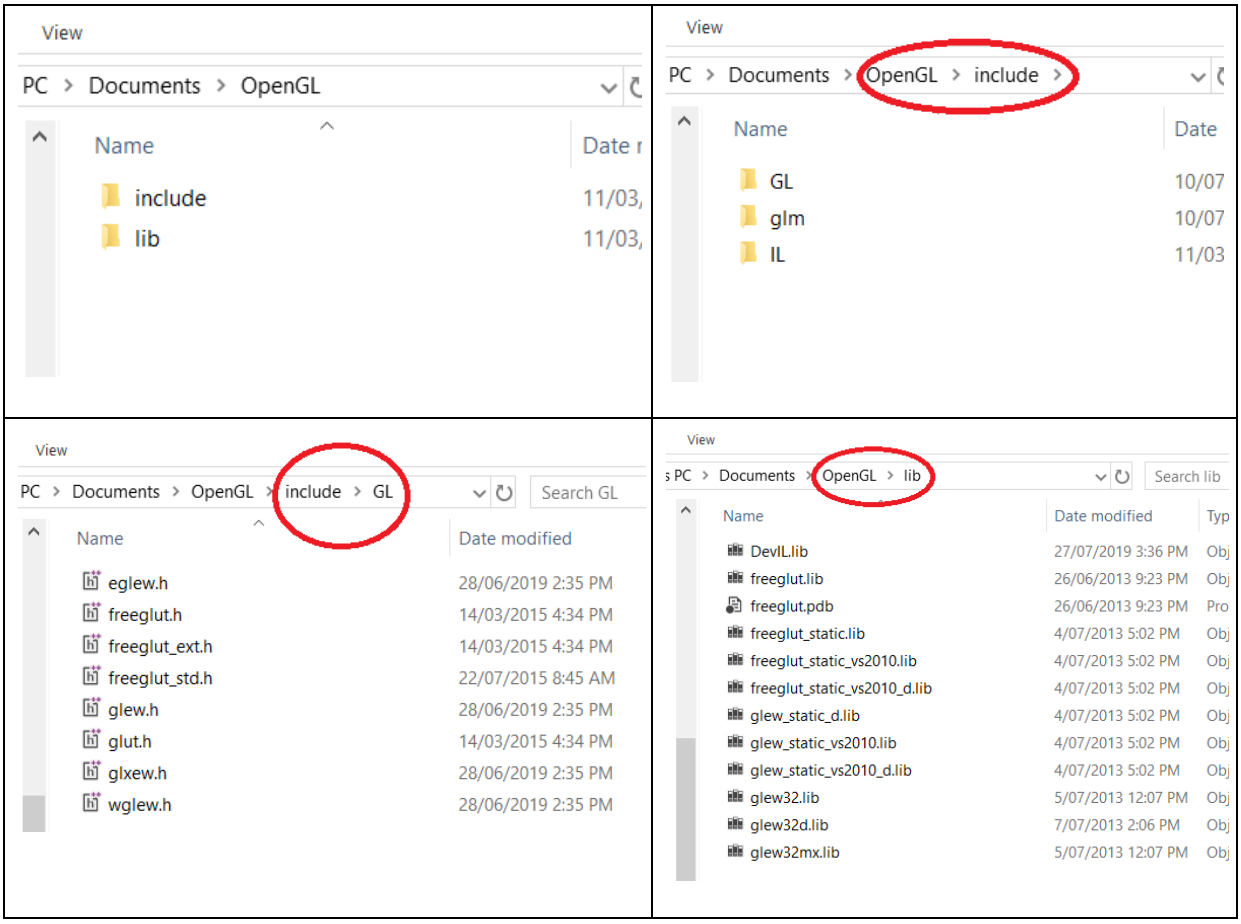
1. Create Project

Start Visual studio and create a new project (**Empty Project**) in a folder of your choice. In this example, the folder is "Documents/MyProject", and the project name is "OpenGLProject".



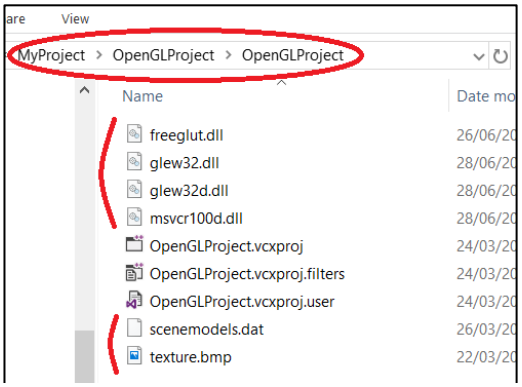
2. OpenGL Setup

- Create another folder "OpenGL" with two subfolders "include" and "lib".
- Copy the contents of "include.zip" to the "include" folder. These files are stored in three subfolders "GL", "glm", and "IL"
- Copy the contents of "lib.zip" to the "lib" folder



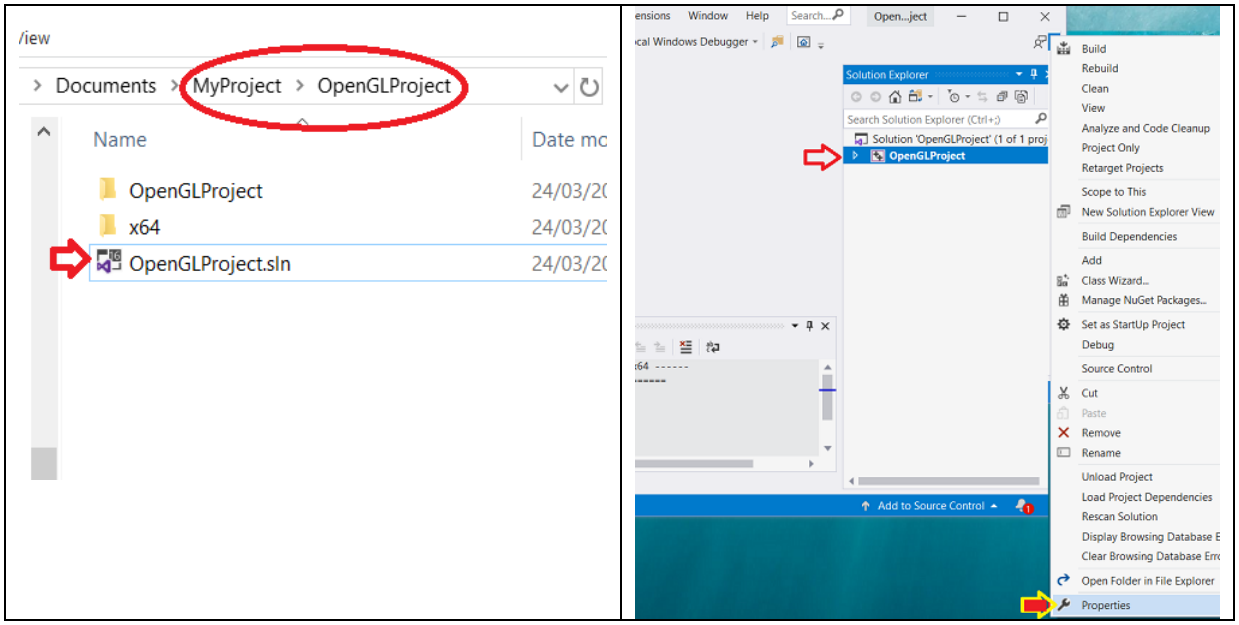
3. Project Setup

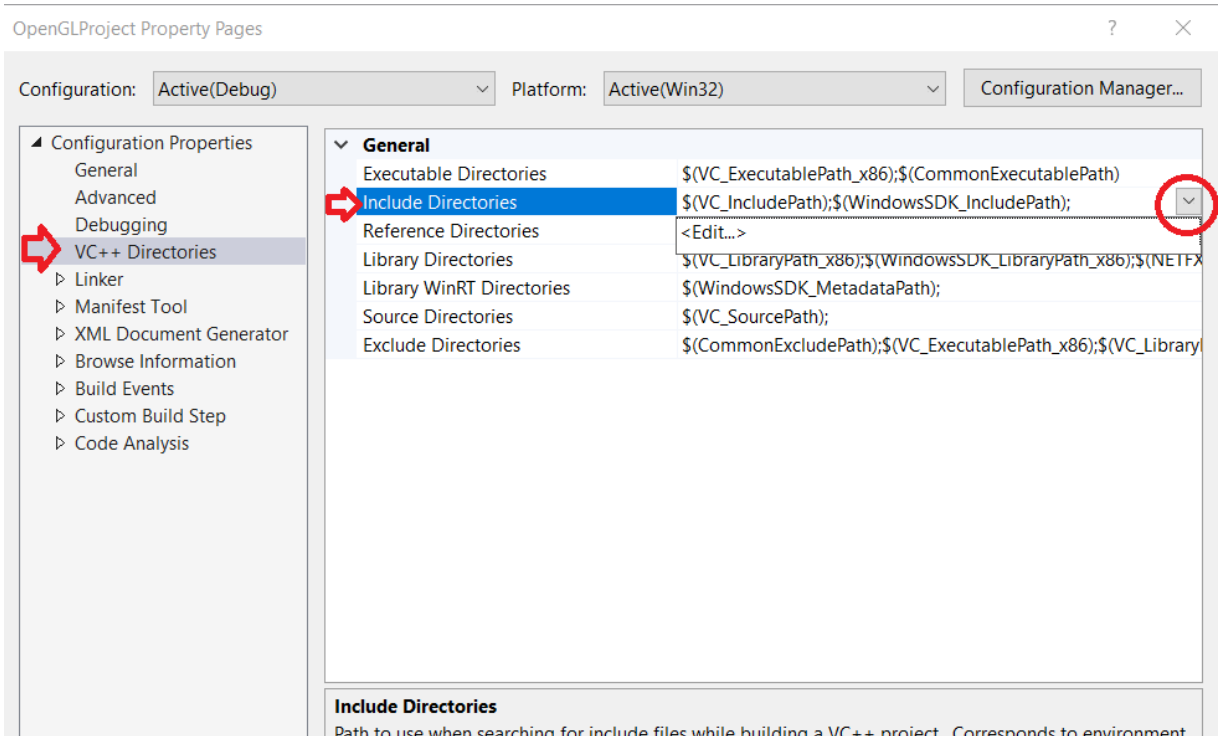
- In Section 1 above, we created a project "OpenGLProject" in the folder "MyProject". This creates a folder "MyProject/OpenGLProject". This folder has a subfolder with the same name: ""MyProject/OpenGLProject/OpenGLProject". This folder is the **project directory**.
- Copy the contents of "dll.zip" to the project directory
- Copy the input files required by your program also to the project directory.

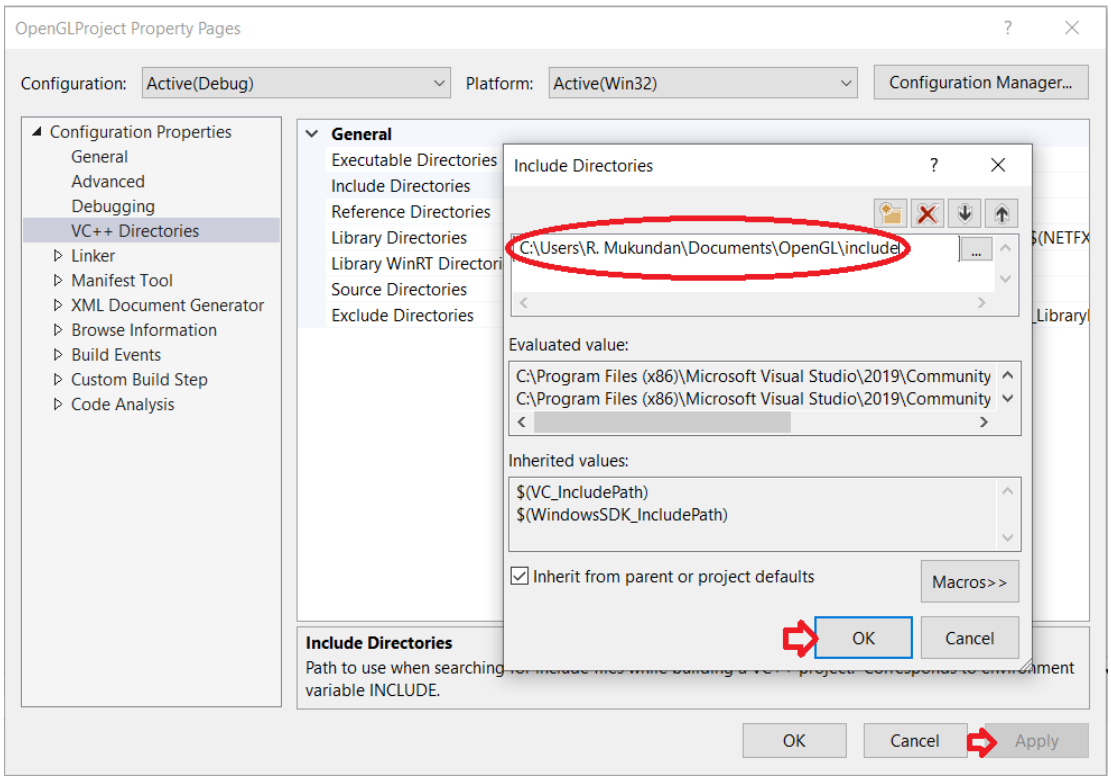


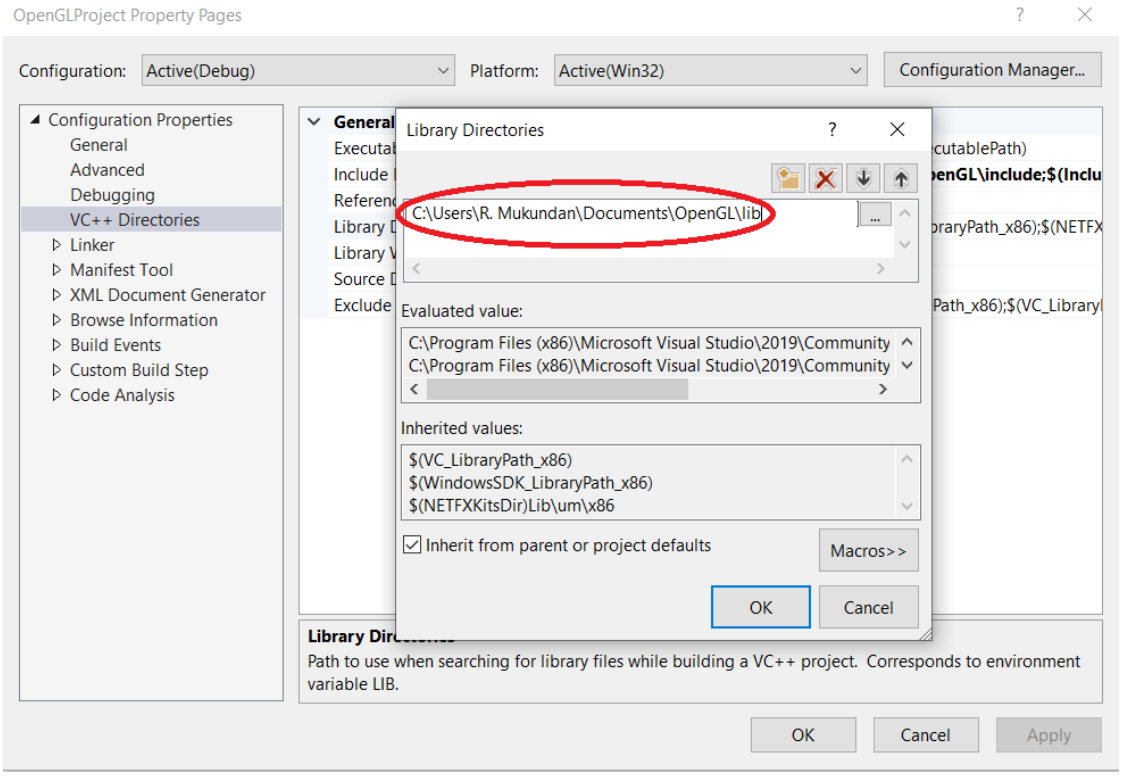
- Go to "MyProject/OpenGLProject" and double-click "OpenGLProject.sln" to launch the project.

- In the “Solution Viewer” window, right click on the project name “OpenGLProject” and select “Properties”
- Select “VC++ Directories” on the left panel of the property pages and select “Include Directories” from the menu. Click the arrow button on the right margin, and select “Edit”.
- Specify the path to the include directory created in Section 2 (see figs. below) and click “OK”.
- In a similar manner, select “Libraries Directories” and specify the path to the library directory created in Section 2.



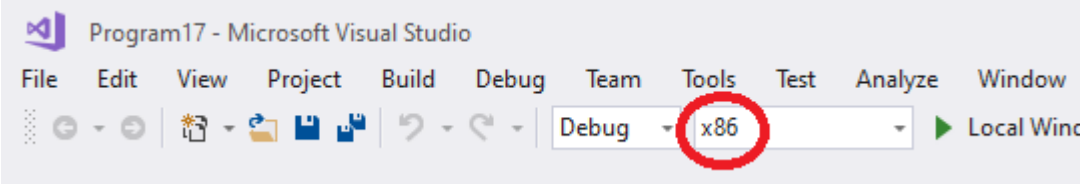






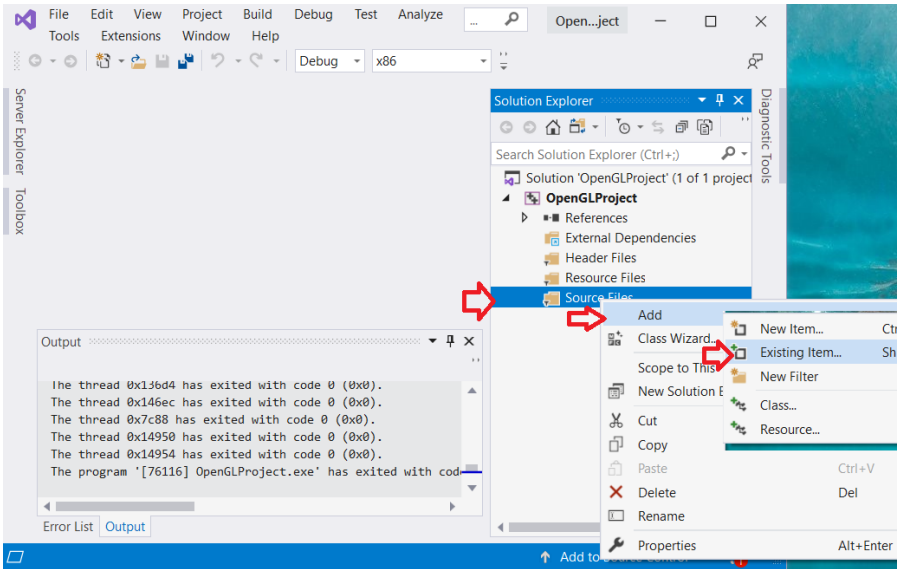
4. Solution Platform

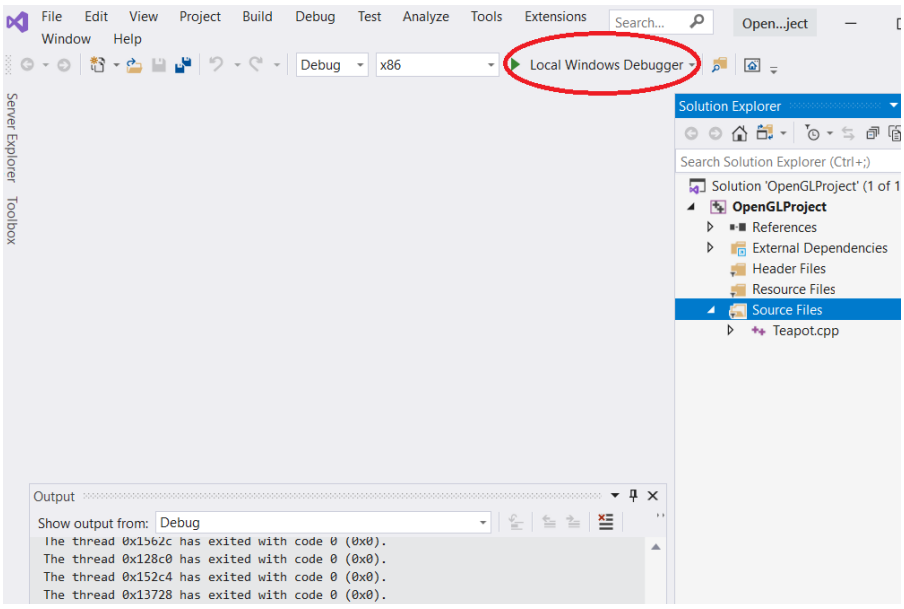
Finally, if the current configuration setting for the solution platform is x64, change it to x86:



5. Running an OpenGL Application

- In the "Solution Explorer" window, right click "Source Files", and select "Add" and then "Existing Item…". Select the program file (.cpp) from your computer.
- Copy all input files (model files, texture files etc.) to the project directory (see Section 3).
- Click the green arrow (Local Windows Debugger) on the menu bar to run the program.





6. Additional step for DevIL, Glew libraries

- COSC363 uses two extra libraries: DevIL for loading textures and GLEW for shader programming using OpenGL-4. The supplied “include”, “lib”, “dll” packs (Sections 2,3) contain the required header, library and dll files. However, one more step is needed to run programs using these libraries:
- On the project properties page (see fig. on Page 3), select “Linker->Input” and click the arrow attached to the “Additional Dependencies” field, and click “Edit…”.
- Input “DevIL.lib” in the text box and click “OK”
- For developing shader applications (OpenGL-4), input “glew32d.lib” in the text box.