Lab01 - Setting up MS Visual Studio 2022 for OpenGL freegut libraries

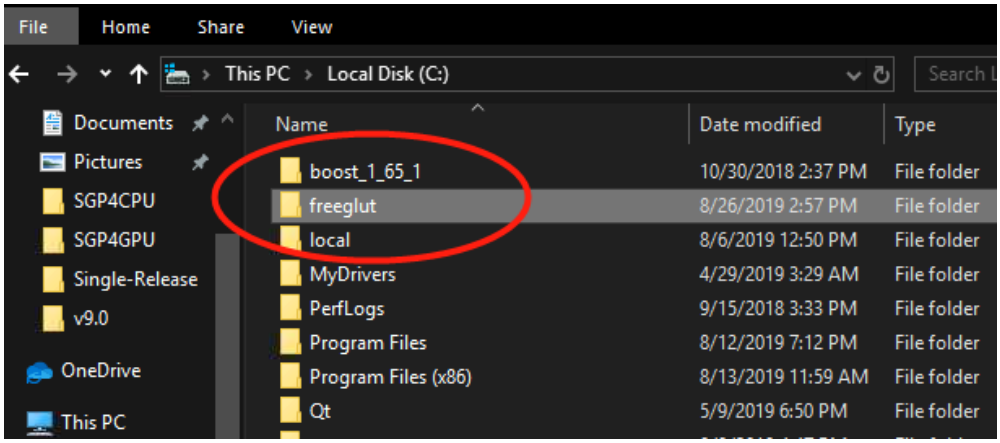
1) Introduction

This is the first of a series of practical sheets that are designed to help you understand how to create 2D and 3D graphic images using C++ and OpenGL graphics library. We will be using the Microsoft Visual Studio 2022 as the IDE (integrated development environment) to enter, save them, compile, link and run computer programs.

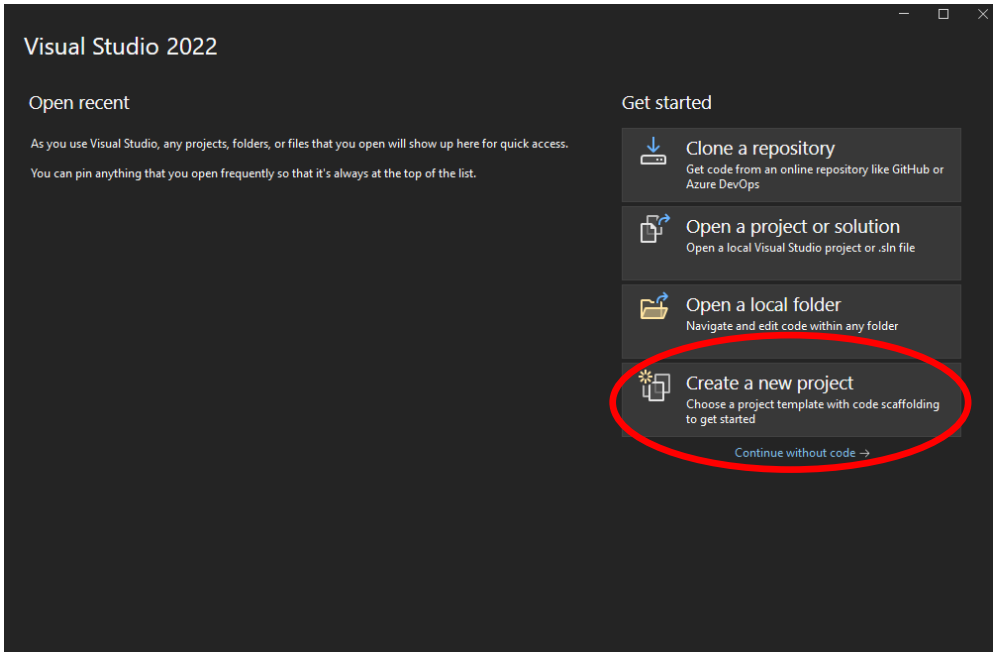
2) Getting started with Microsoft Visual Studio 2022 and OpenGL graphics library

The following steps get you started with your very first OpenGL program using Microsoft Visual Studio 2022.

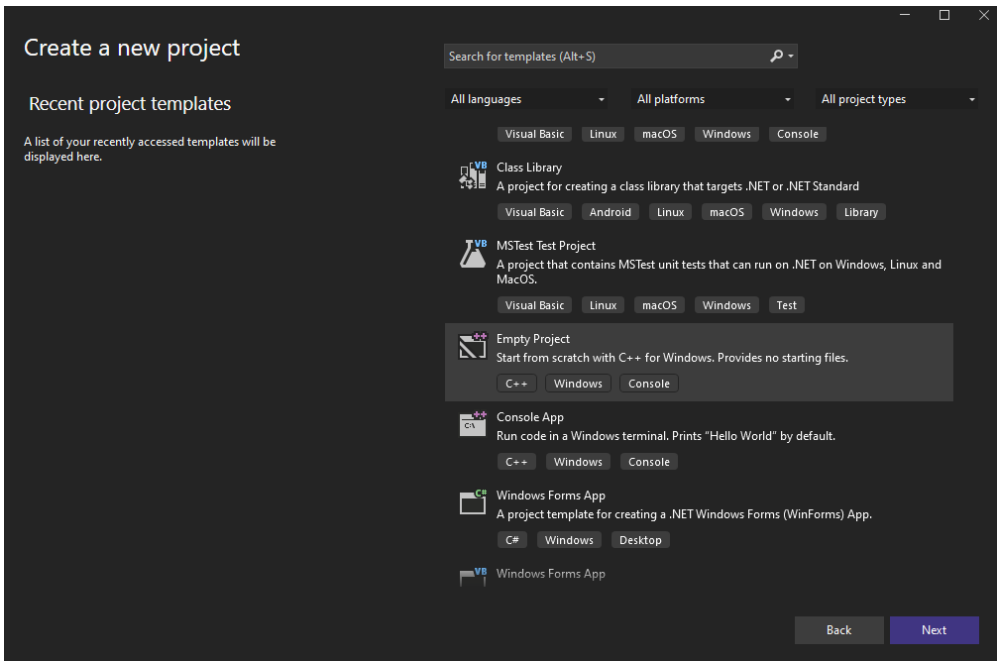
1. Uncompress the file named 'freeglut.rar' to your local disk C, D and so on. The freeglut graphics library is for Windows 10 x64 and Visual Studio 2022.



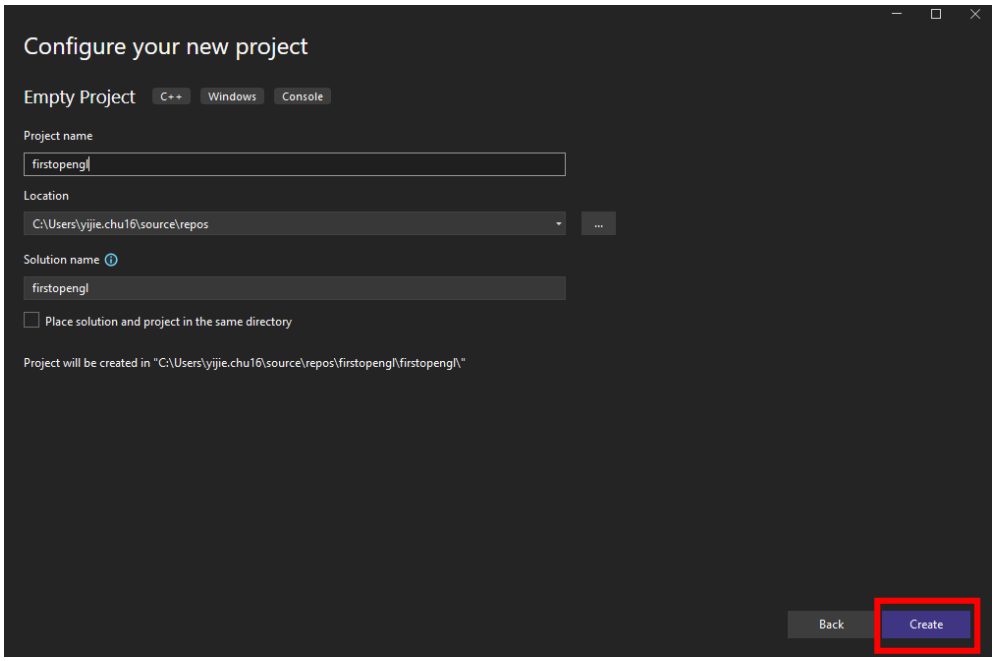
2. Open Microsoft Visual Studio 2022, and you will see the following window (note that this may take a couple of minutes).



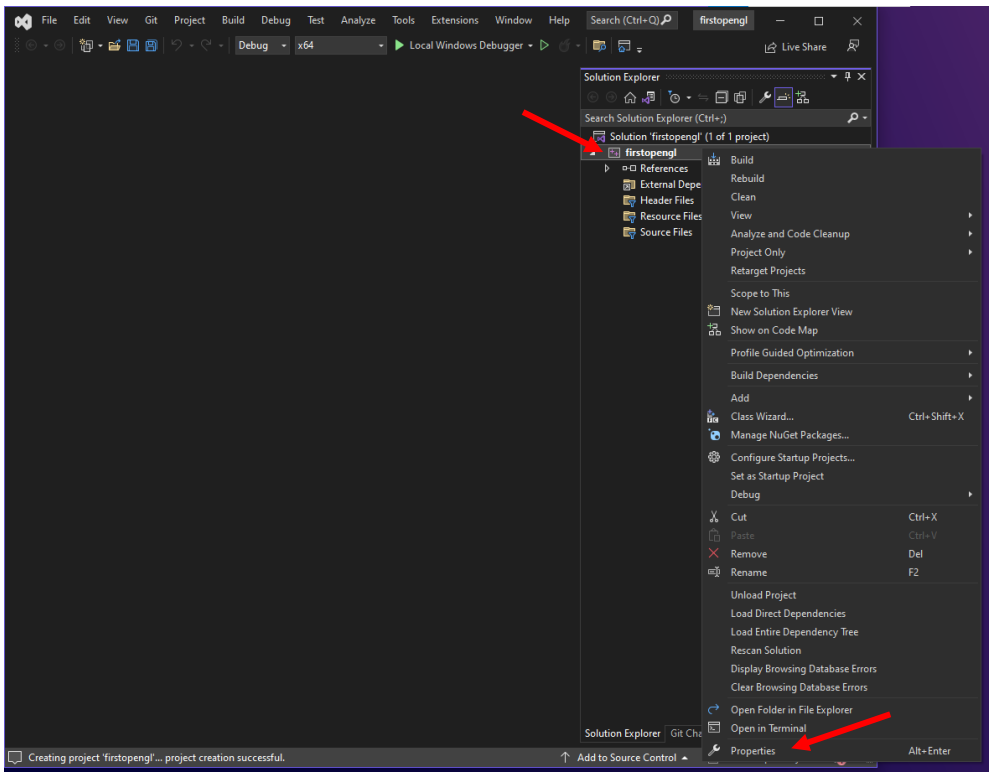
3. Click 'Create a new project', and you will see the following window. Choose 'Empty Project' and click 'Next'.



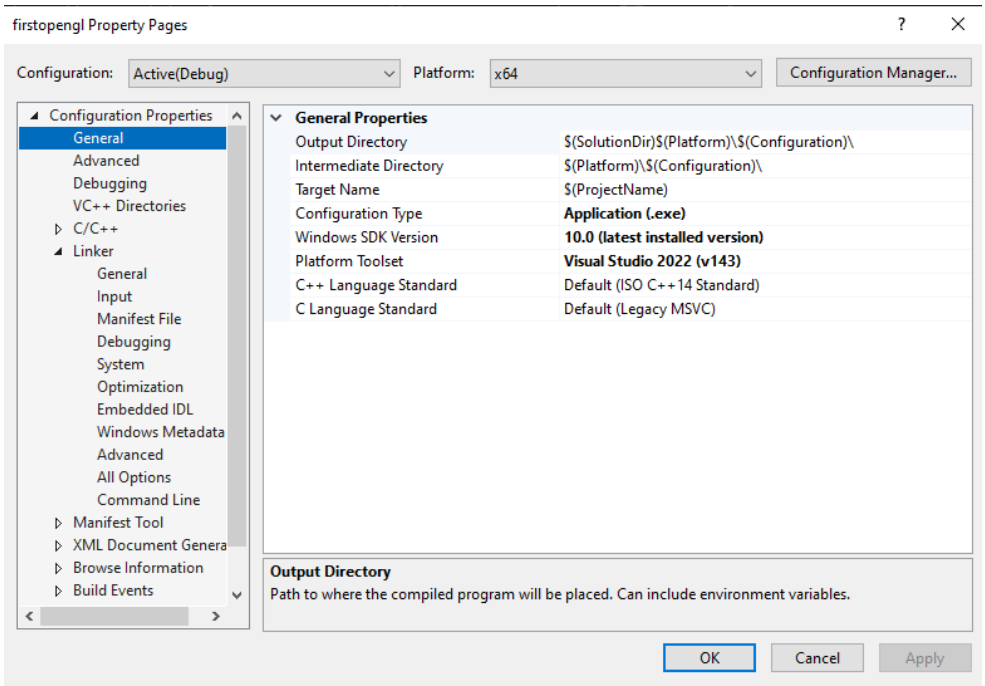
4. Enter a name for the project such as that in the following picture, ‘firstopengl’. Then click ‘Create’.



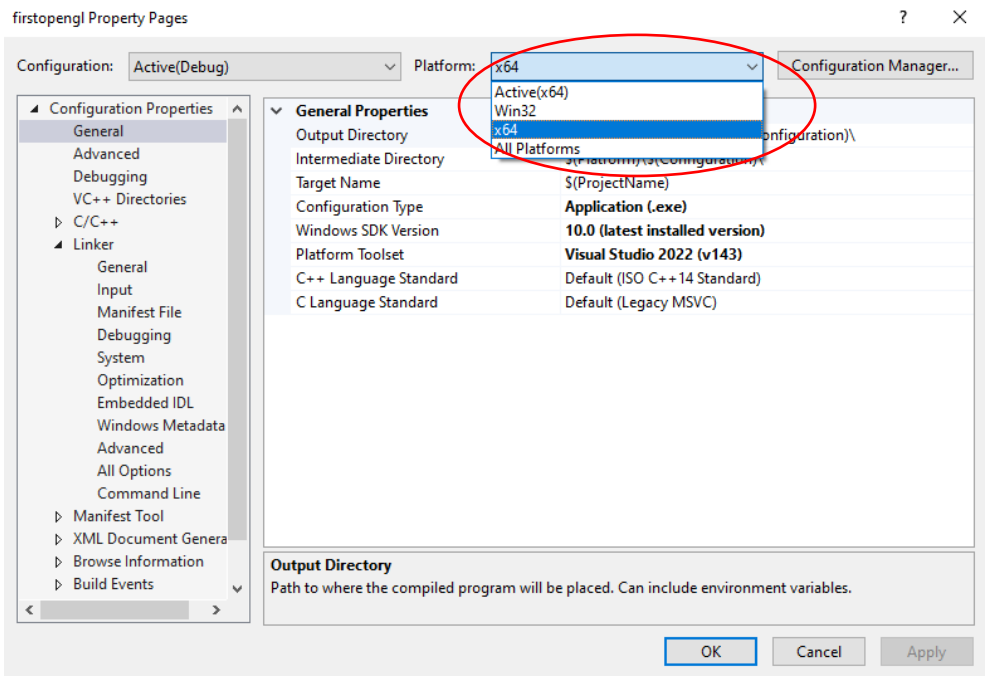
5. Now right click on the project name, and choose properties (in the bottom of the list).



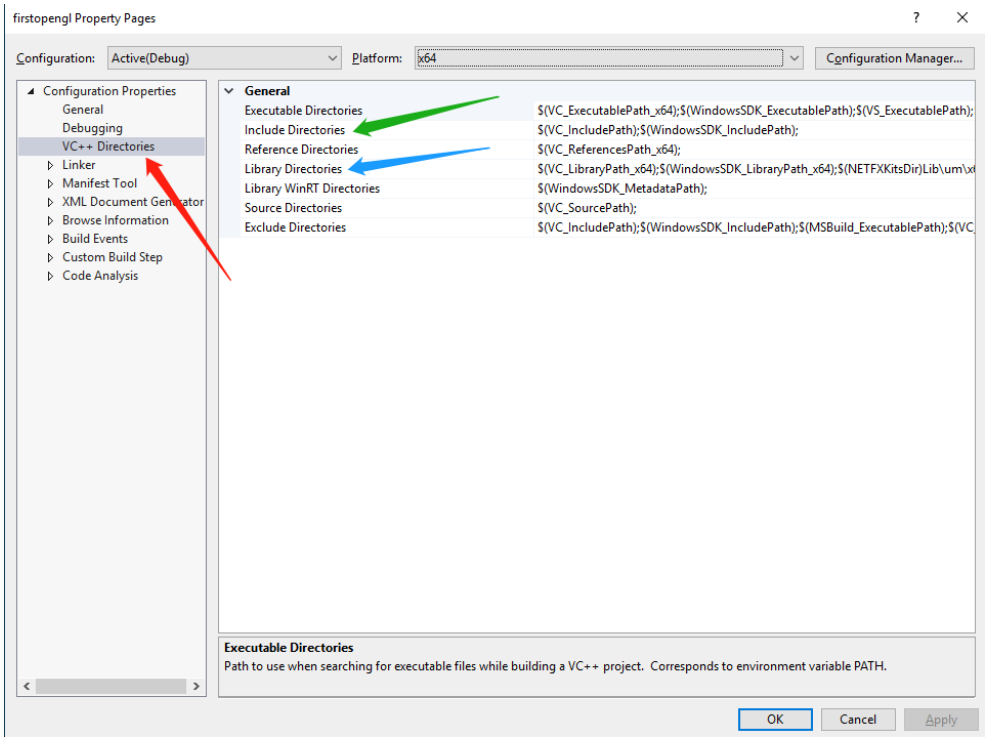
6. Then you will see the following.



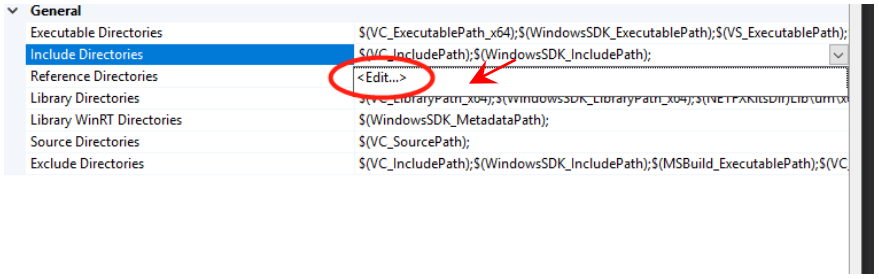
7. Click the symbol ‘v’ beside the ‘Platform’, choose ‘x64’ in the list under ‘Platform’.



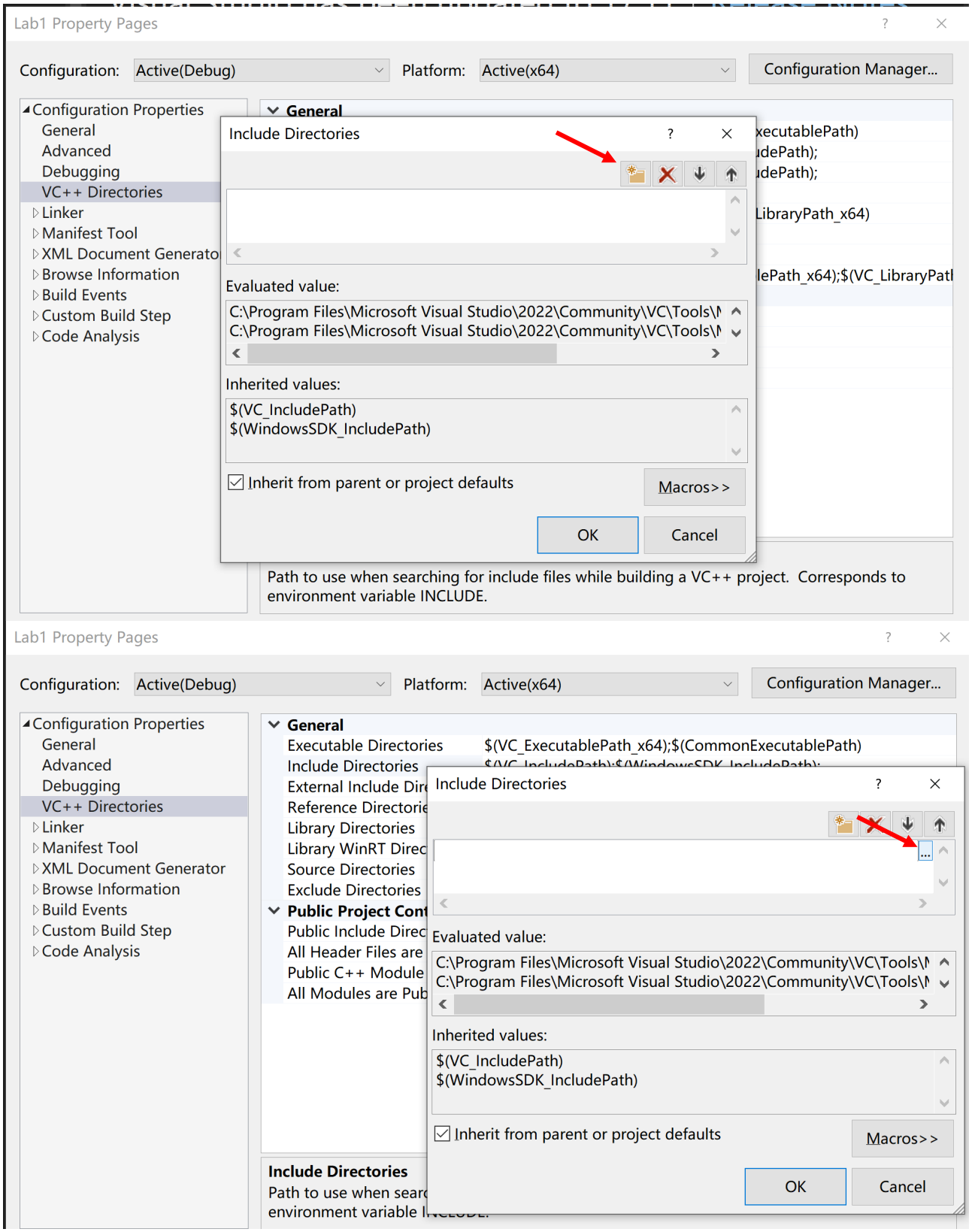
8. Click ‘VC++ Directories’ on the left side (as shown with the red arrow), and you will see the following picture.

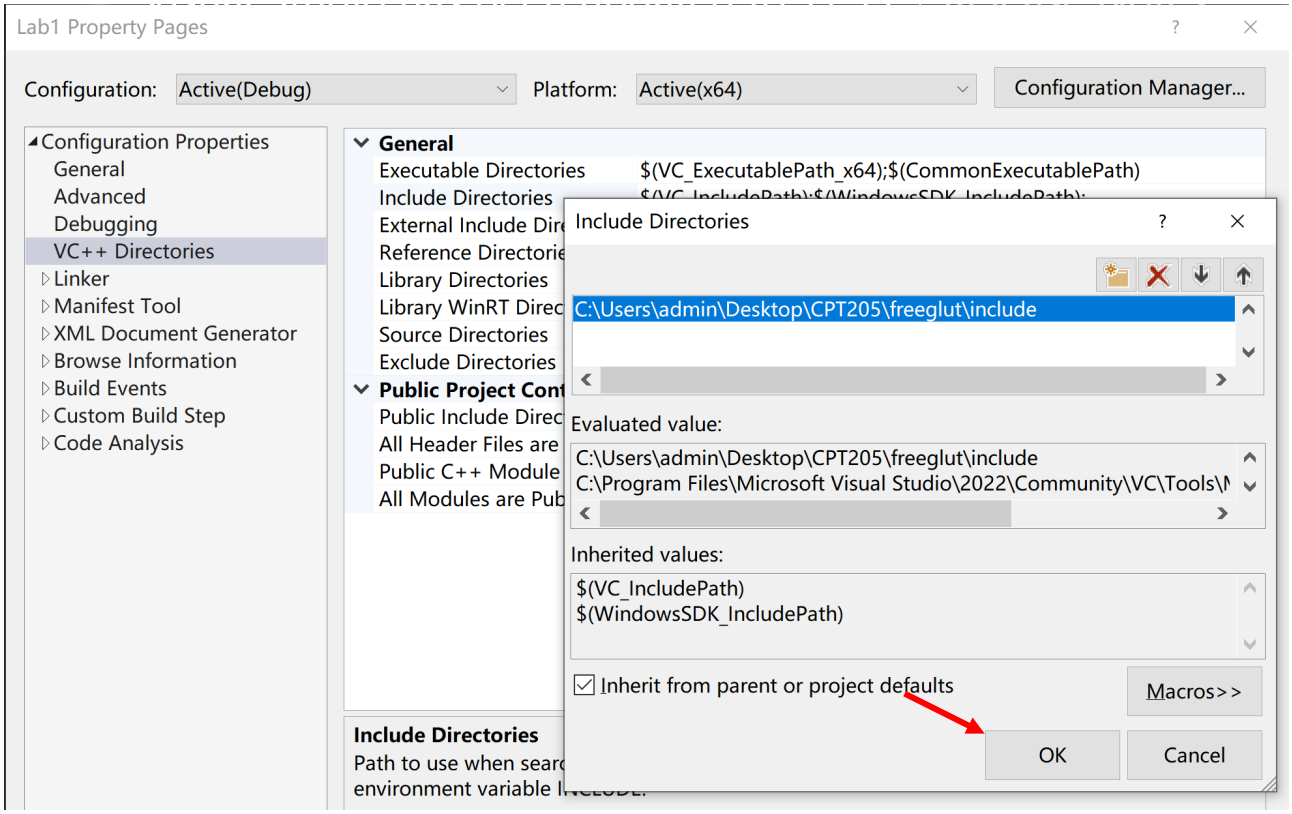


9. Click the symbol ‘v’ which is on the right side of ‘Include Directories’, and choose the ‘<Edit...>’ in the list.

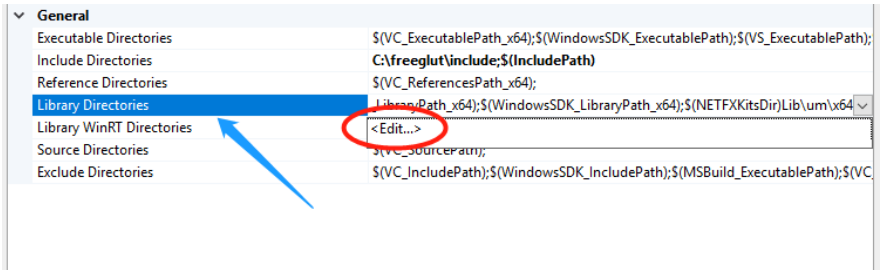


10. Click the ‘New Line’ button (red arrow) and choose the ‘include’ folder where you have uncompressed the zip file. Click the ‘OK’ button, to go back to the previous page.

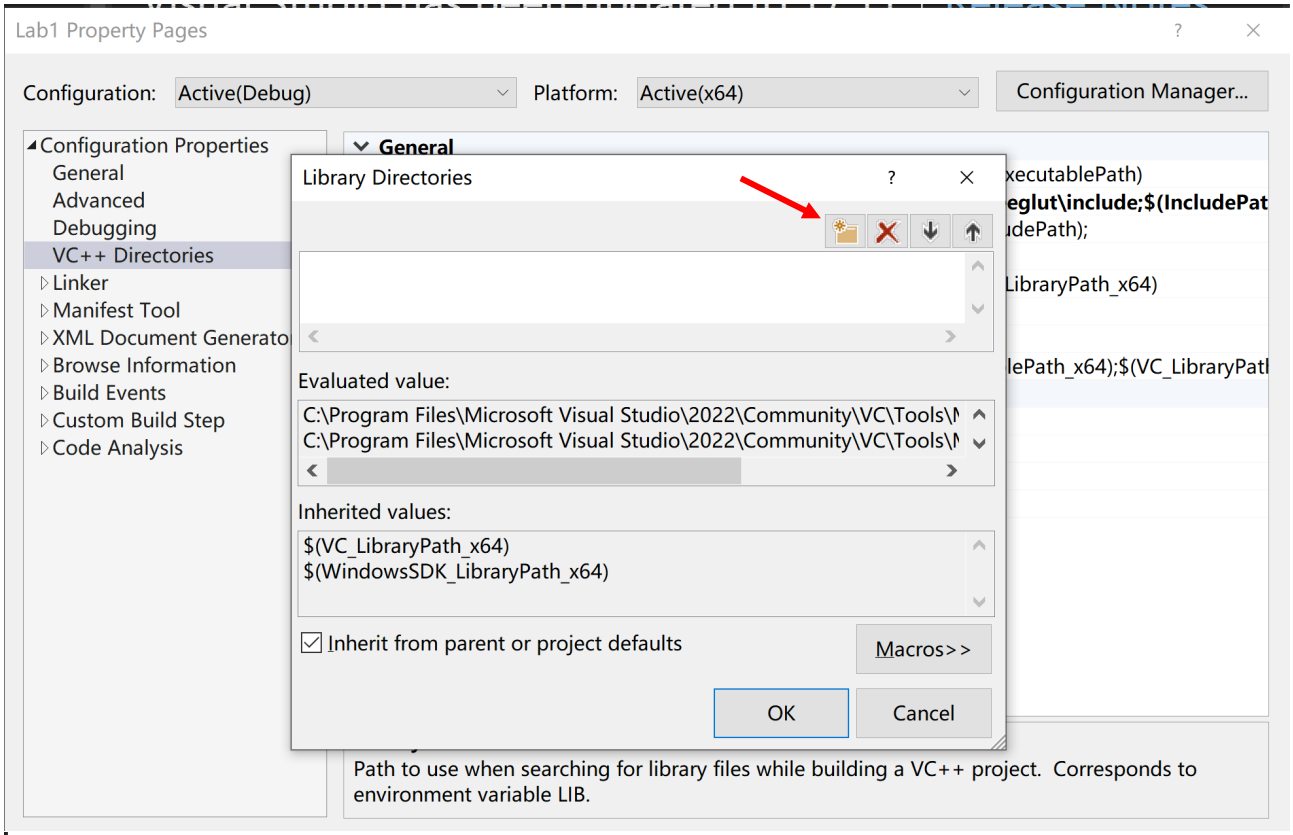


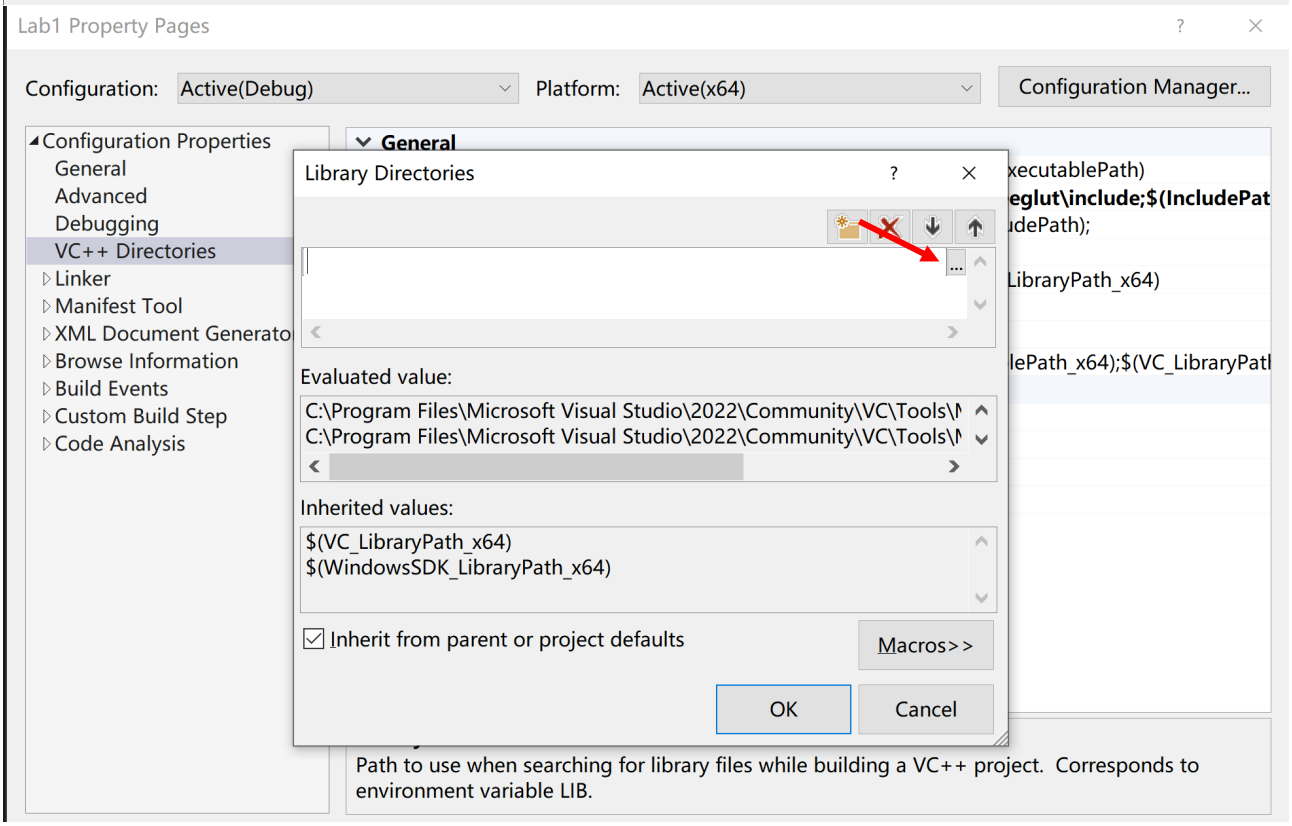


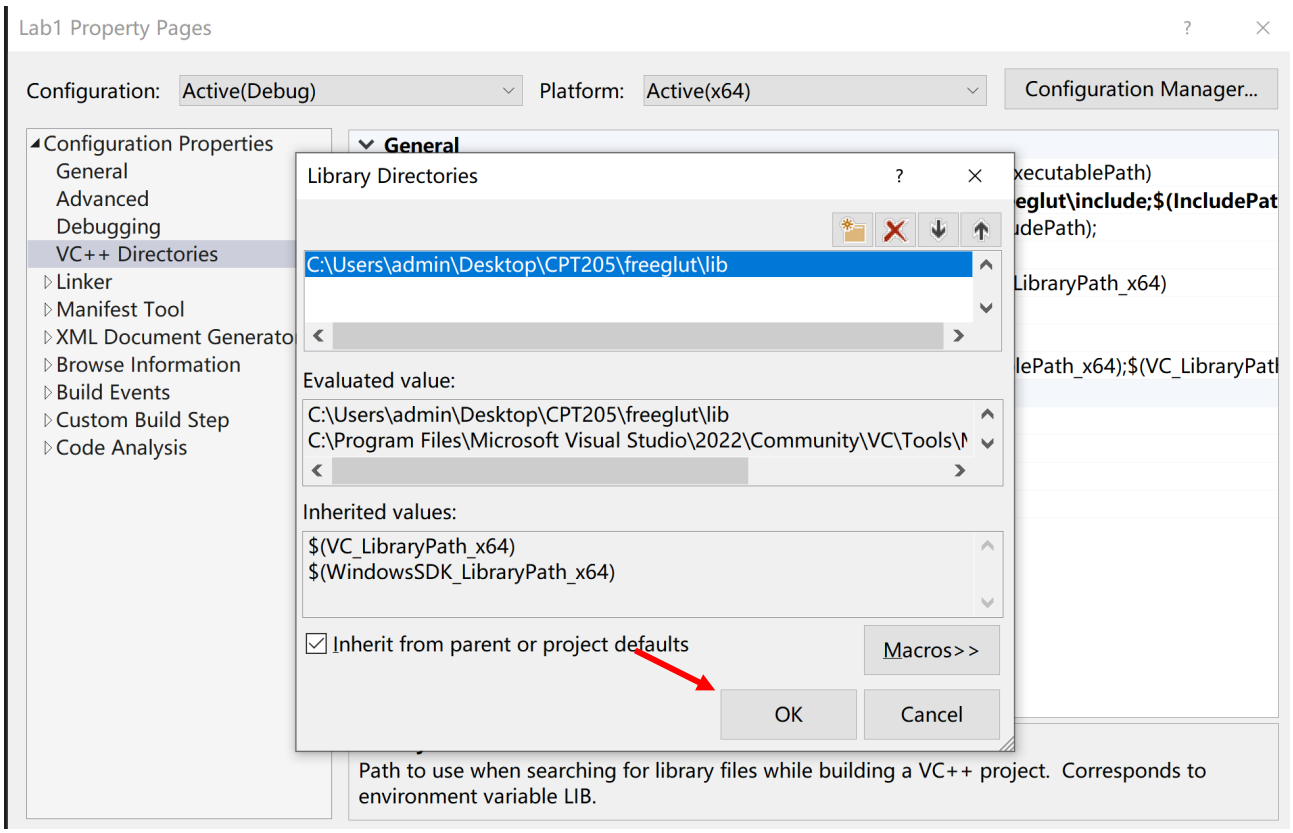
11. Click symbol ‘v’ on the right side of ‘Library Directories’ (blue arrow), and choose ‘<Edit...>’ in the list.



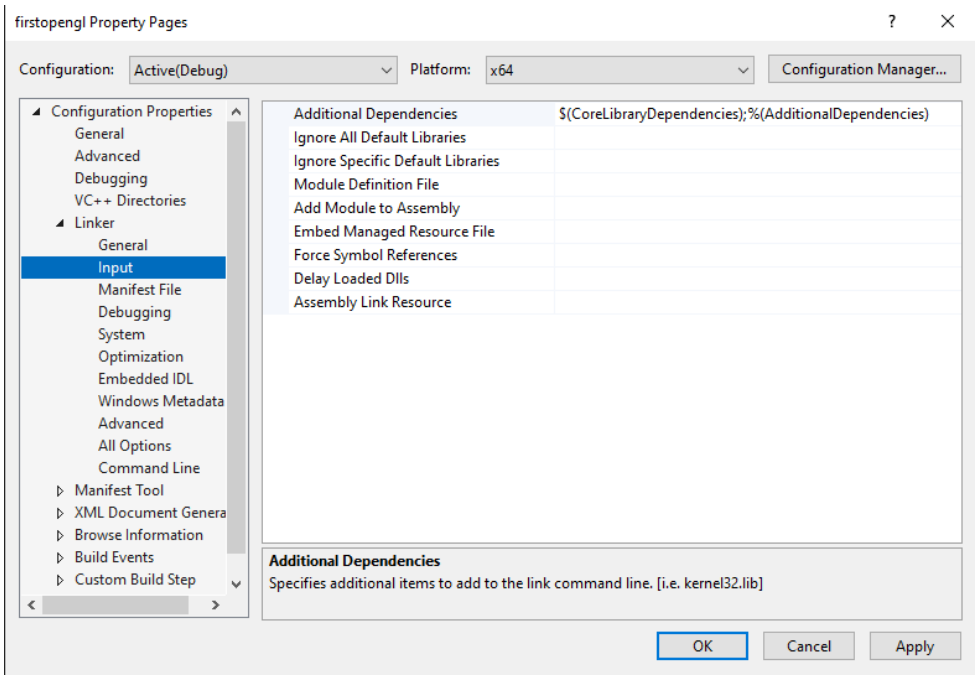
12. Click the ‘New Line’ button (red arrow), and choose the ‘lib’ folder where you have uncompressed the rar file. Click the ‘OK’ button, to go back to the previous page.


Lab1 Property Pages
Configuration: Active(Debug)
Platform: Active(x64)
Configuration Manager...
Configuration Properties
General
Advanced
Debugging
VC++ Directories
Linker
Manifest Tool
XML Document Generato
Browse Information
Build Events
Custom Build Step
Code Analysis
General
Library Directories
Evaluated value:
C:\Program Files\Microsoft Visual Studio\2022\Community\VC\Tools\M
C:\Program Files\Microsoft Visual Studio\2022\Community\VC\Tools\M
Inherited values:
$(VC_LibraryPath_x64)
$(WindowsSDK_LibraryPath_x64)
Inherit from parent or project defaults
Macros>>
OK
Cancel
Path to use when searching for library files while building a VC++ project. Corresponds to environment variable LIB.

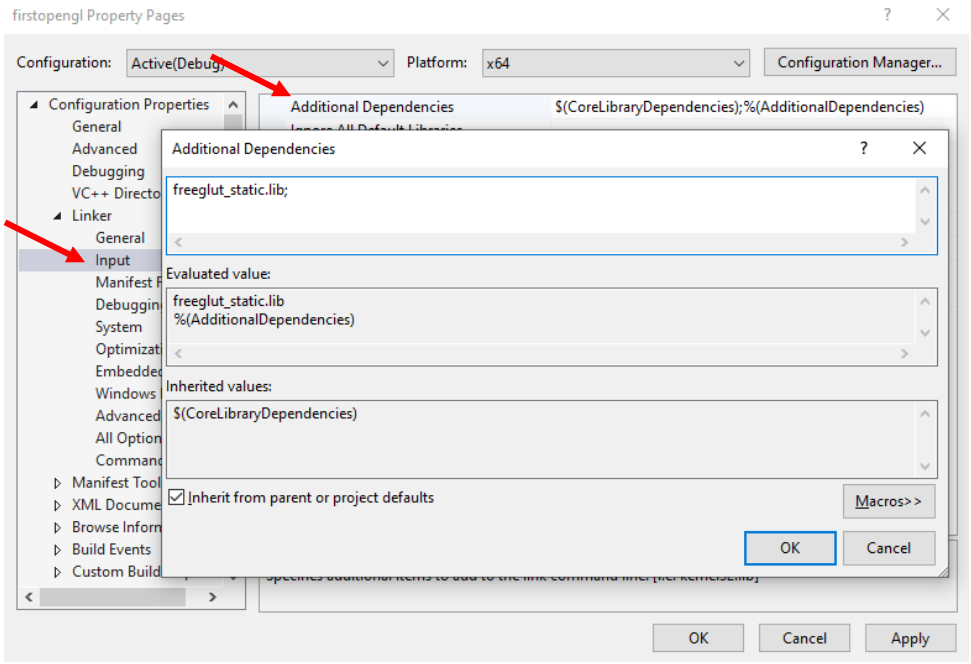

Lab1 Property Pages
Configuration: Active(Debug)
Platform: Active(x64)
Configuration Manager...
Configuration Properties
General
Advanced
Debugging
VC++ Directories
Linker
Manifest Tool
XML Document Generato
Browse Information
Build Events
Custom Build Step
Code Analysis
General
Library Directories
Evaluated value:
C:\Program Files\Microsoft Visual Studio\2022\Community\VC\Tools\M
C:\Program Files\Microsoft Visual Studio\2022\Community\VC\Tools\M
Inherited values:
$(VC_LibraryPath_x64)
$(WindowsSDK_LibraryPath_x64)
Inherit from parent or project defaults
Macros>>
OK
Cancel
Path to use when searching for library files while building a VC++ project. Corresponds to environment variable LIB.



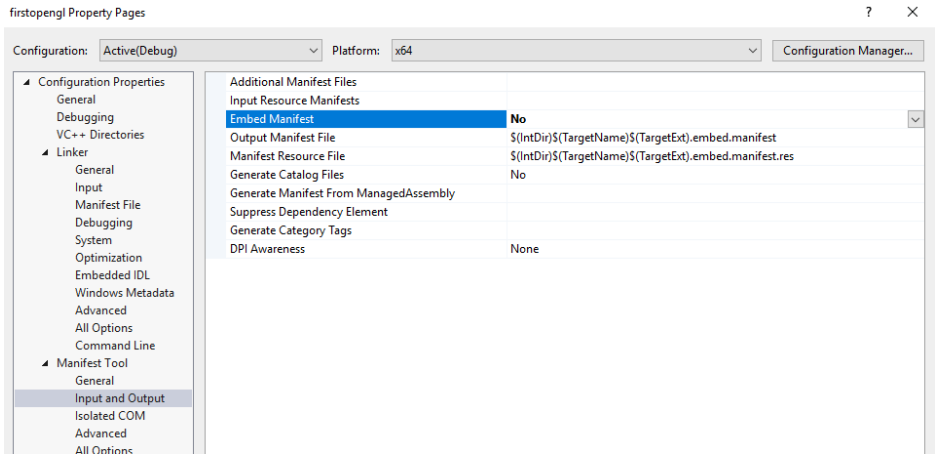
13. Click the symbol '>' which is on the left side of 'Linker', and choose 'Input' in the list of 'Linker'; you will see the following picture.



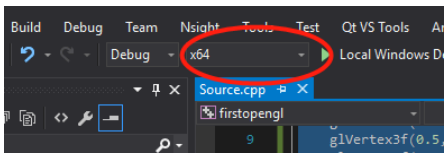
14. Add the text "freeglut_static.lib;" into 'Additional Dependencies' which is on the top of the right part of the current window. A semicolon ';' is used to separate the library



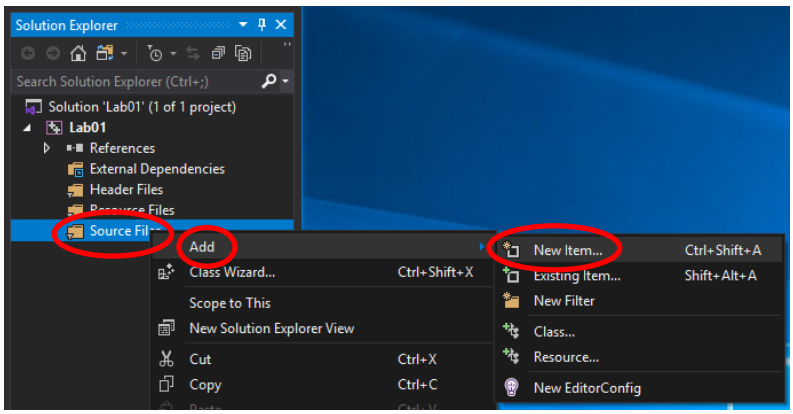
15. Click the '>' symbol which is on the left side of 'Manifest Tool', and choose 'Input and Output' in the list of 'Manifest Tool'; Set 'Embed Manifest' to 'No'; Click 'OK'.



16. Choose 'x64' in the Platform list



17. Now you are ready to write and edit your program. We have a sample program for you to use. Right click 'Source Files' (in 'Solution Explorer' in the right of the window), then click 'Add' and 'New Item...', and add a 'C++ File (.cpp)' file. Copy the following code into the cpp file. Make sure that you do not copy unwanted characters (e.g. page numbers when copy the code across multiple pages) which will result in errors on compilation.

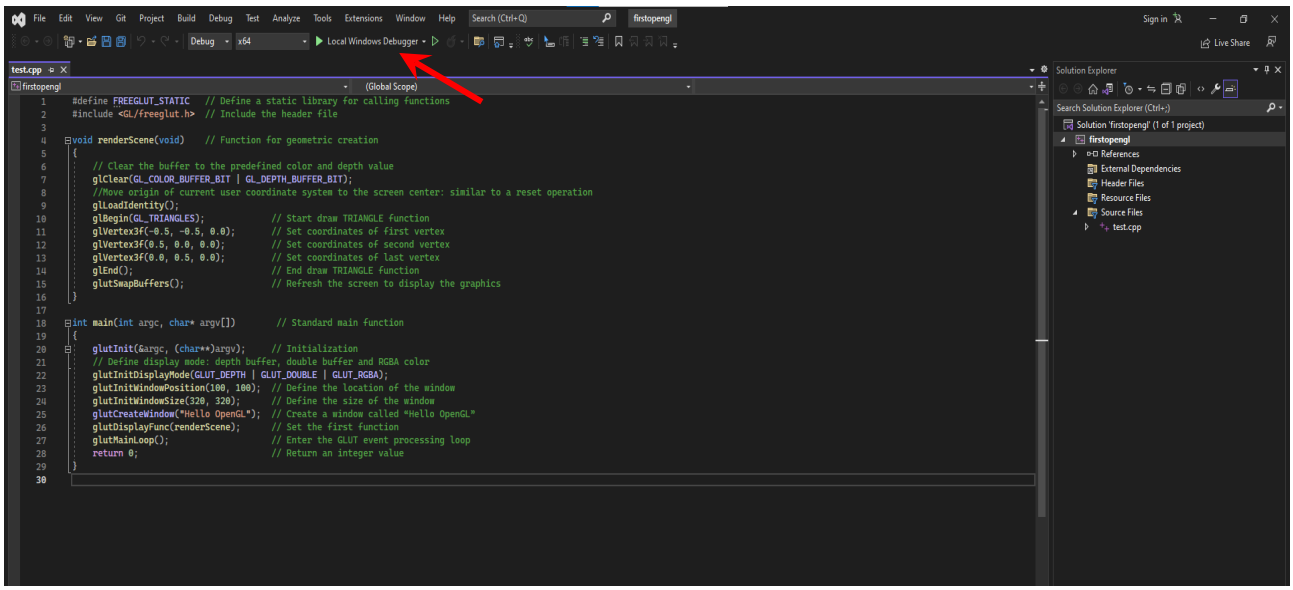



```
#define FREEGLUT_STATIC   // Define a static library for calling functions
#include <GL/freeglut.h>  // Include the header file

void renderScene(void)    // Function for geometric creation
{
     // Clear the buffer to the predefined color and depth value
     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
     //Move origin of current user coordinate system to the screen center: similar to a reset operation
     glLoadIdentity();
     glBegin(GL_TRIANGLES);              // Start draw TRIANGLE function
     glVertex3f(-0.5, -0.5, 0.0);        // Set coordinates of first vertex
     glVertex3f(0.5, 0.0, 0.0);          // Set coordinates of second vertex
     glVertex3f(0.0, 0.5, 0.0);          // Set coordinates of last vertex
     glEnd();                            // End draw TRIANGLE function
     glutSwapBuffers();                  // Refresh the screen to display the graphics
}

int main(int argc, char* argv[])         // Standard main function
{
     glutInit(&argc, (char**)argv);      // Initialization
     // Define display mode: depth buffer, double buffer and RGBA color
     glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA);
     glutInitWindowPosition(100, 100);  // Define the location of the window
     glutInitWindowSize(320, 320);       // Define the size of the window
     glutCreateWindow("Hello OpenGL");  // Create a window called "Hello OpenGL"
     glutDisplayFunc(renderScene);       // Set the first function
     glutMainLoop();                     // Enter the GLUT event processing loop
     return 0;                           // Return an integer value
}
```

18. Press 'F7' on your keyboard to build the source code, then press 'F5' to run the program now; Or just click the compile button (green one).



19. If you have done it correctly, a window with a triangle will come out.