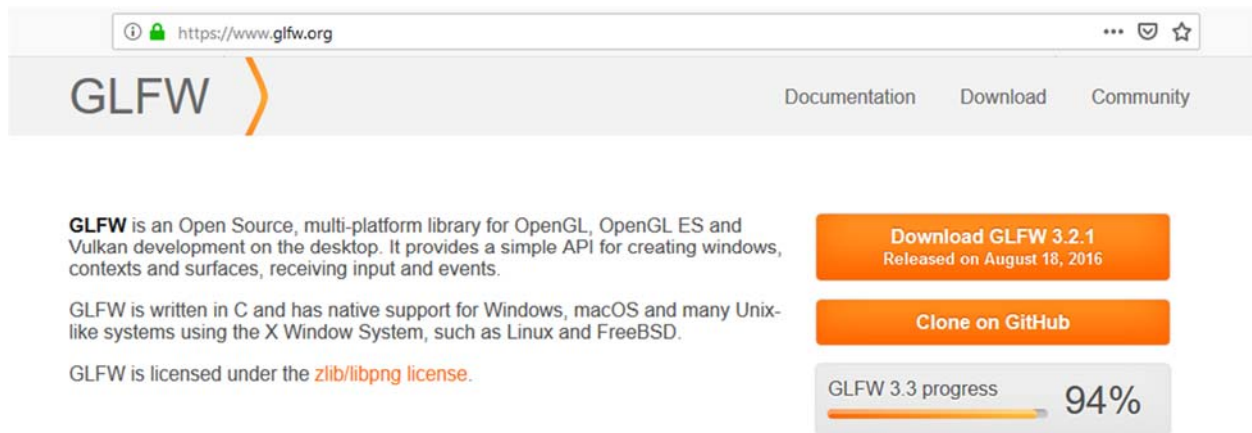


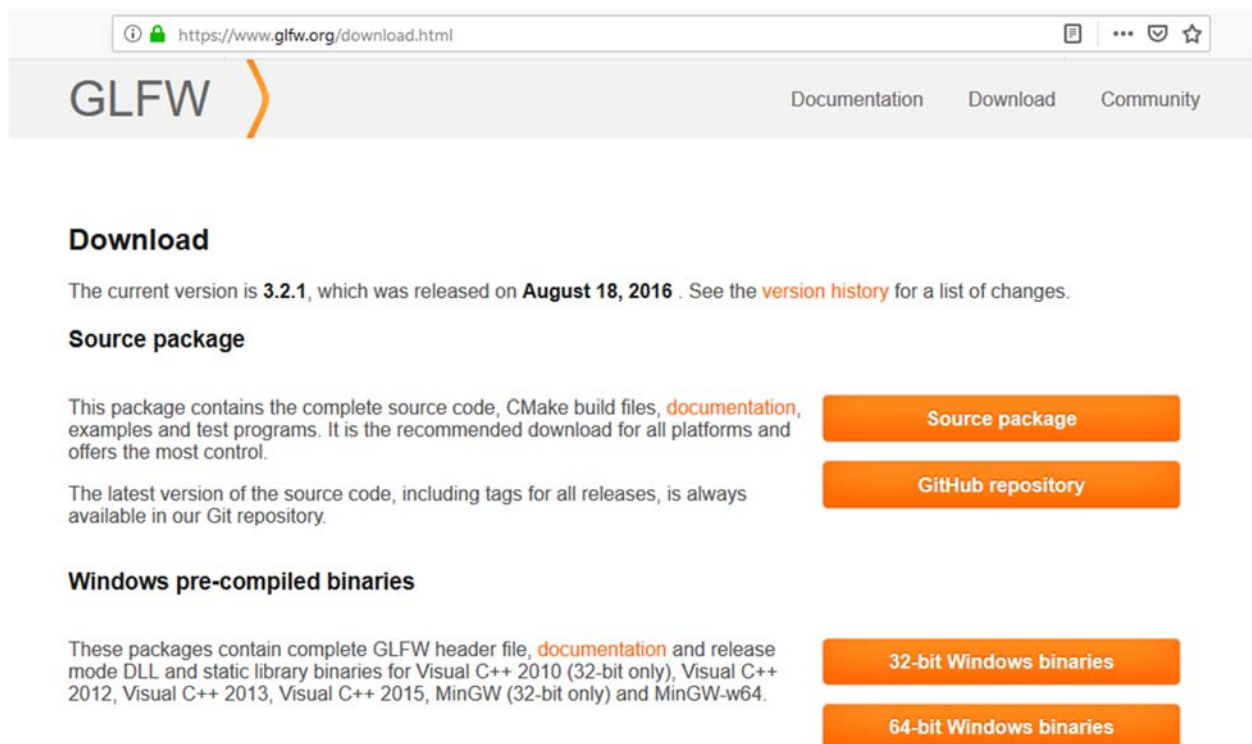
إعداد GLFW

يمكن تحميل المكتبة من الموقع

[/https://www.glfw.org](https://www.glfw.org)

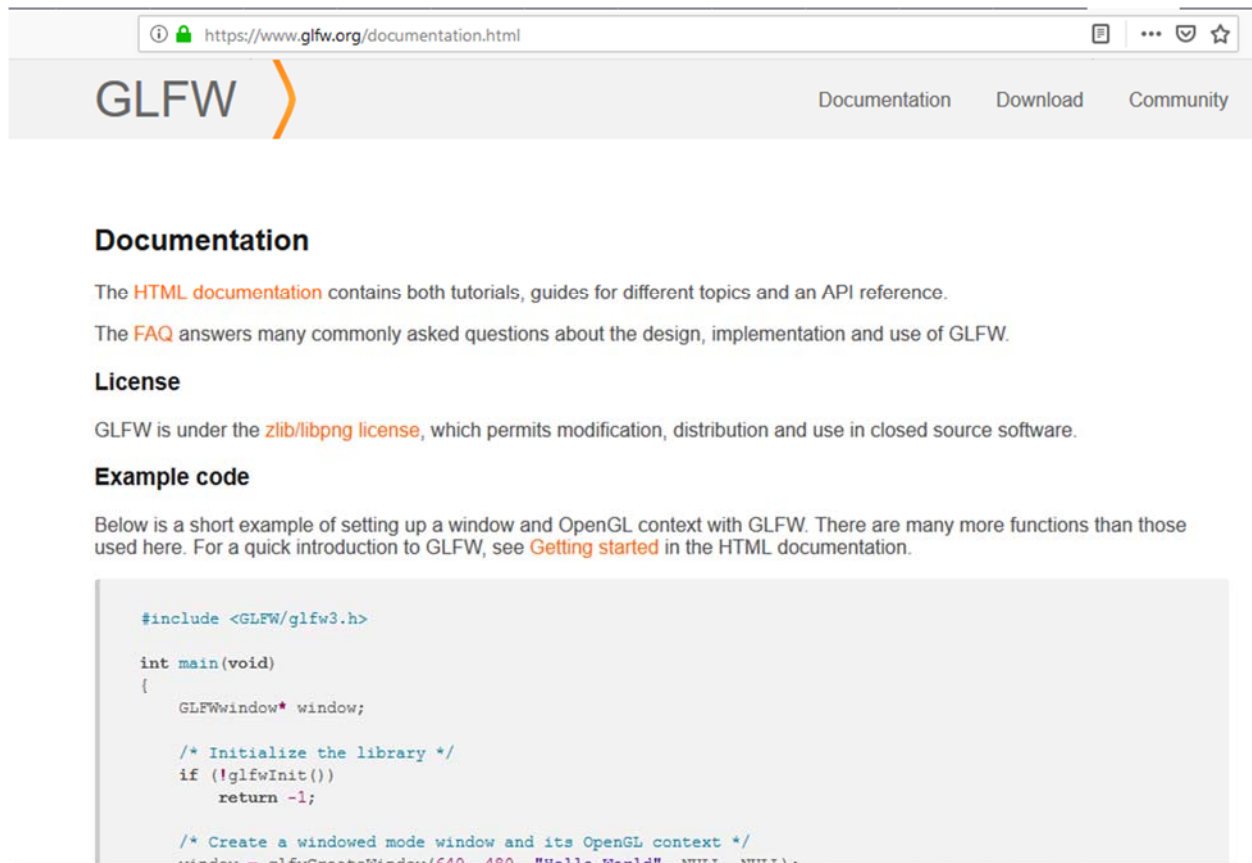
The screenshot shows the GLFW website homepage. The browser address bar displays "https://www.glfw.org". The website header includes the GLFW logo and navigation links for "Documentation", "Download", and "Community". The main content area describes GLFW as an Open Source, multi-platform library for OpenGL, OpenGL ES, and Vulkan development. It provides a simple API for creating windows, contexts, and surfaces, receiving input and events. Below this, it states that GLFW is written in C and has native support for Windows, macOS, and many Unix-like systems using the X Window System, such as Linux and FreeBSD. It also mentions that GLFW is licensed under the zlib/libpng license. On the right side, there are two orange buttons: "Download GLFW 3.2.1" (Released on August 18, 2016) and "Clone on GitHub". Below these buttons is a progress bar for "GLFW 3.3 progress" showing 94% completion.

من الرابط Download



The screenshot shows the GLFW download page. The browser address bar displays "https://www.glfw.org/download.html". The website header includes the GLFW logo and navigation links for "Documentation", "Download", and "Community". The main content area is titled "Download" and states that the current version is 3.2.1, which was released on August 18, 2016. It also mentions that users can see the version history for a list of changes. Below this, there is a section titled "Source package" which describes the package as containing the complete source code, CMake build files, documentation, examples, and test programs. It is the recommended download for all platforms and offers the most control. The latest version of the source code, including tags for all releases, is always available in their Git repository. To the right of this text are two orange buttons: "Source package" and "GitHub repository". Below the "Source package" section is a section titled "Windows pre-compiled binaries" which describes packages containing complete GLFW header files, documentation, and release mode DLL and static library binaries for Visual C++ 2010 (32-bit only), Visual C++ 2012, Visual C++ 2013, Visual C++ 2015, MinGW (32-bit only), and MinGW-w64. To the right of this text are two orange buttons: "32-bit Windows binaries" and "64-bit Windows binaries".

من الرابط Documentation يمكن الوصول إلى تعليمات الاستخدام وبعض الأمثلة:



**Documentation**

The [HTML documentation](#) contains both tutorials, guides for different topics and an API reference.

The [FAQ](#) answers many commonly asked questions about the design, implementation and use of GLFW.

**License**

GLFW is under the [zlib/libpng license](#), which permits modification, distribution and use in closed source software.

**Example code**

Below is a short example of setting up a window and OpenGL context with GLFW. There are many more functions than those used here. For a quick introduction to GLFW, see [Getting started](#) in the HTML documentation.

```
#include <GLFW/glfw3.h>

int main(void)
{
    GLFWwindow* window;

    /* Initialize the library */
    if (!glfwInit())
        return -1;

    /* Create a windowed mode window and its OpenGL context */
    window = glfwCreateWindow(640, 480, "Hello World", NULL, NULL);
```

مثال لتشغيل المكتبة:

```
#include <GLFW/glfw3.h>

int main(void)
{
    GLFWwindow* window;

    /* Initialize the library */
    if (!glfwInit())
        return -1;

    /* Create a windowed mode window and its OpenGL context */
    window = glfwCreateWindow(640, 480, "Hello World", NULL, NULL);
    if (!window)
    {
        glfwTerminate();
        return -1;
    }

    /* Make the window's context current */
    glfwMakeContextCurrent(window);

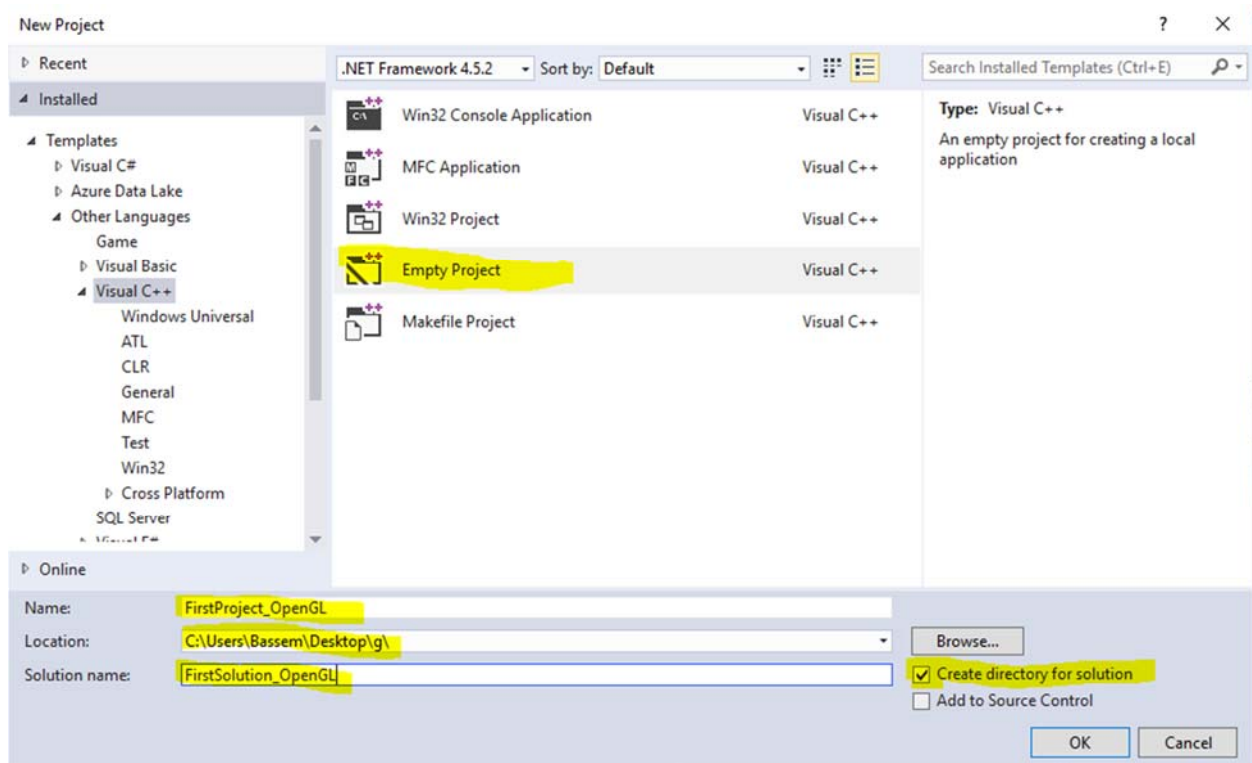
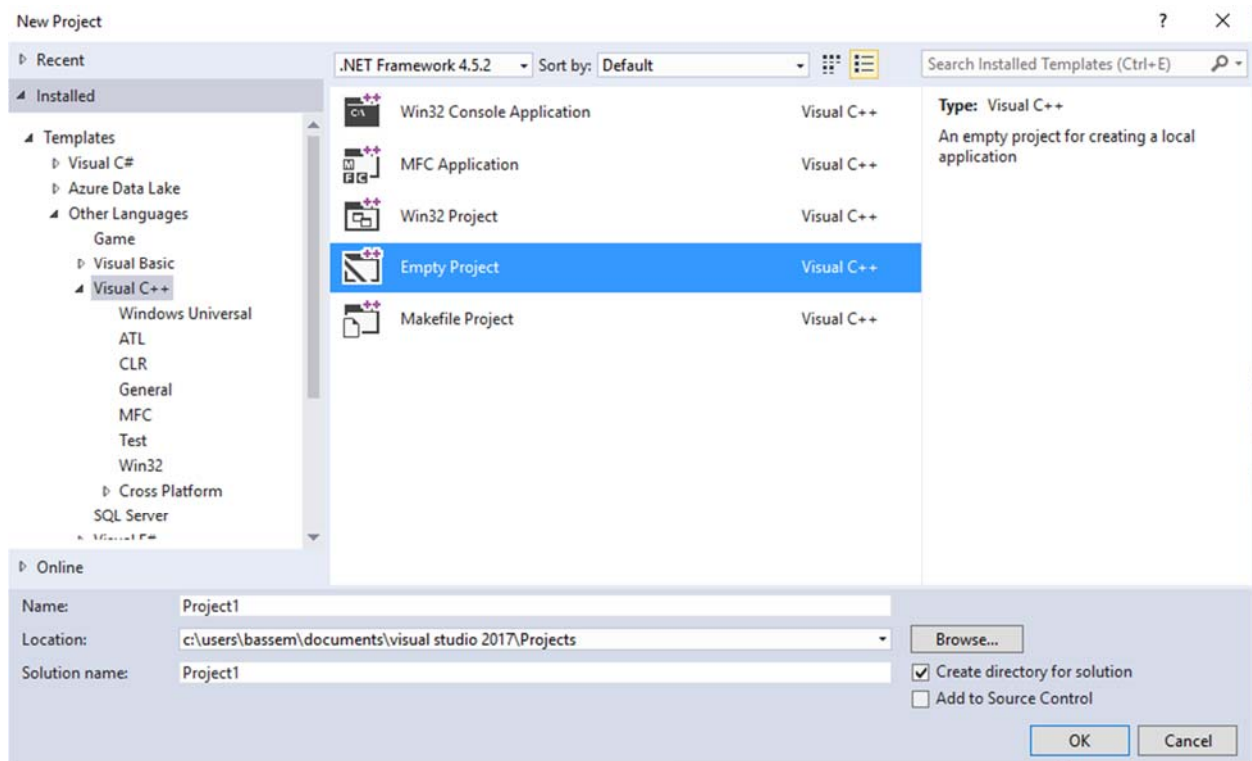
    /* Loop until the user closes the window */
    while (!glfwWindowShouldClose(window))
    {
        /* Render here */
        glClear(GL_COLOR_BUFFER_BIT);

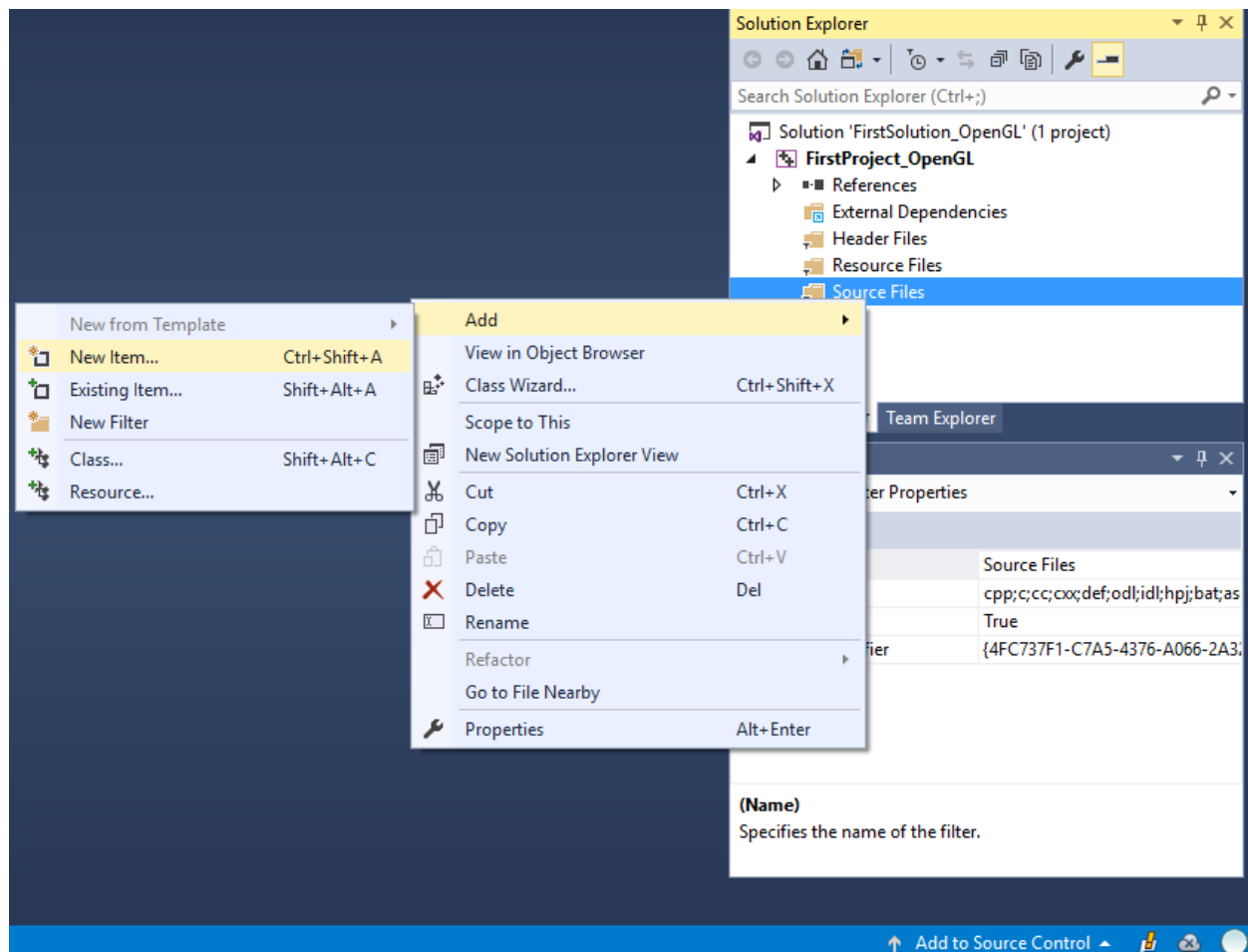
        /* Swap front and back buffers */
        glfwSwapBuffers(window);

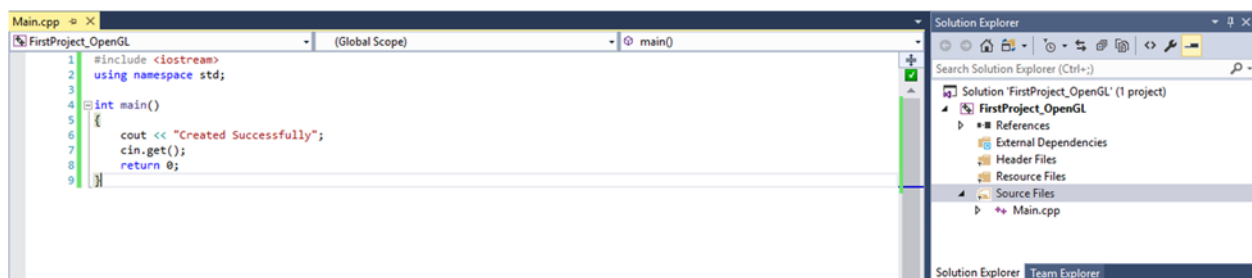
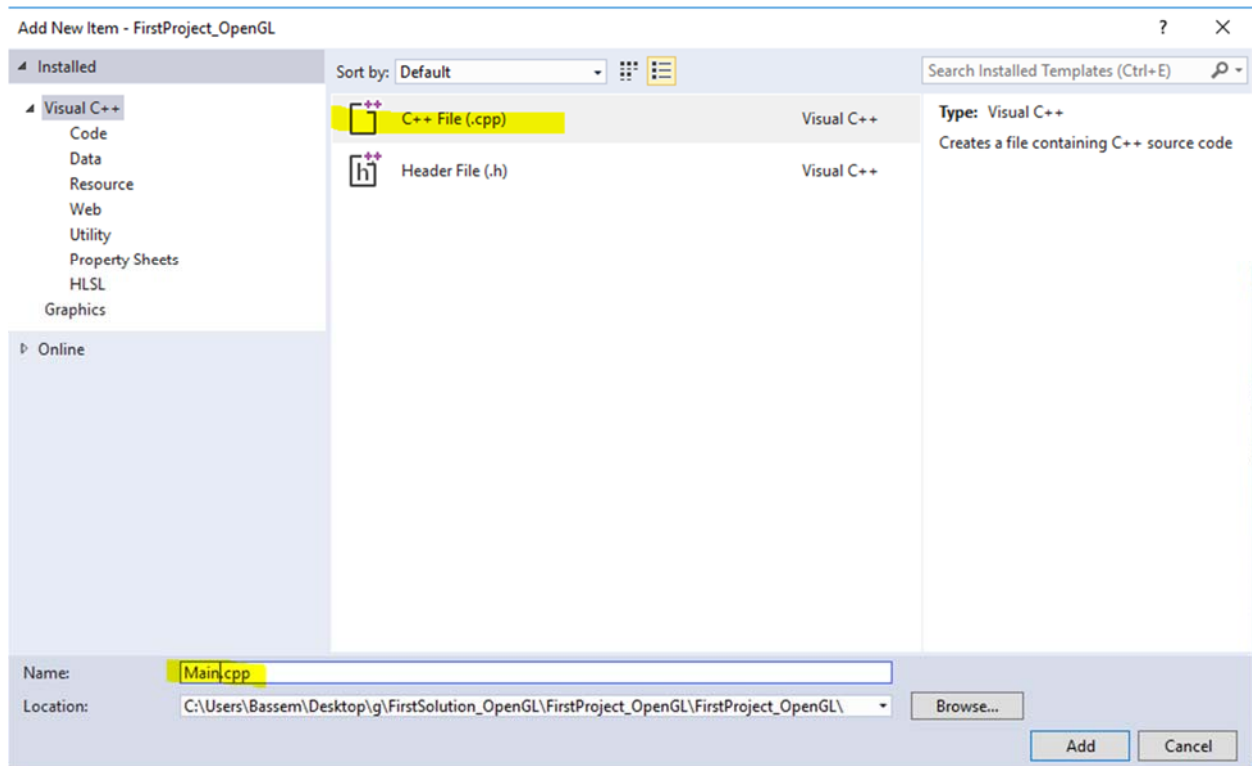
        /* Poll for and process events */
        glfwPollEvents();
    }

    glfwTerminate();
    return 0;
}
```

## أولا إنشاء مشروع C++:





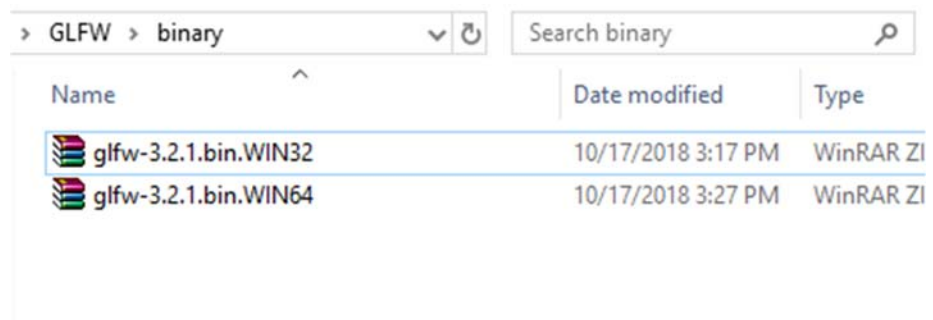


## ثانياً تضمين المكتبة:

اولاً تحميل المكتبة من الموقع السابق (مرفقة في المجلد المزود للطلاب في مجلد GLFW)

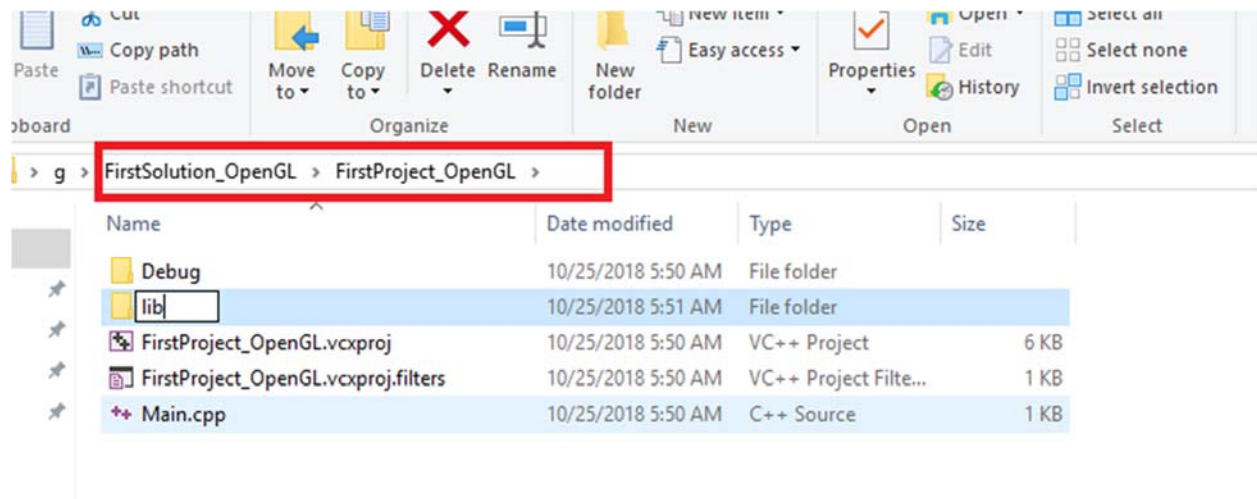
في حال العمل من المجلد الموجود مع الطلبة:

من المسار GLFW\binary

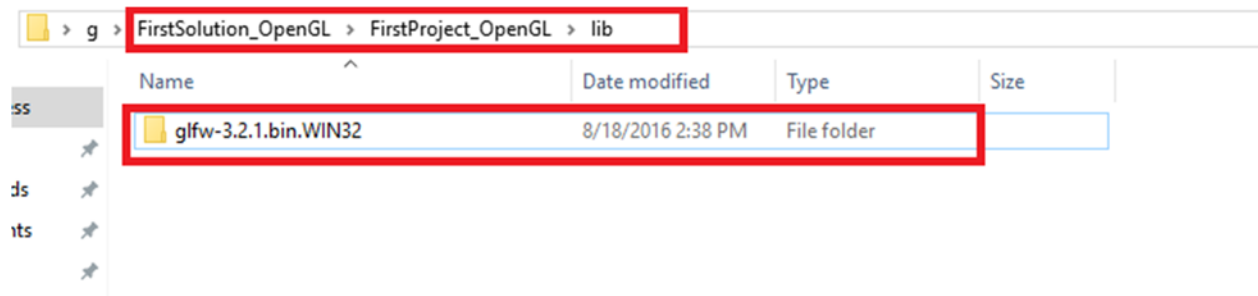


حسب نوع المعالج الذي نحتاج تشغيل التطبيق عليه نختار احد الملفين المضغوطين ونقوم بفك ضغطه (غالباً WIN32)

في مجلد المشروع الذي تم إنشاؤه نقوم بإنشاء مجلد جديد وليكن lib



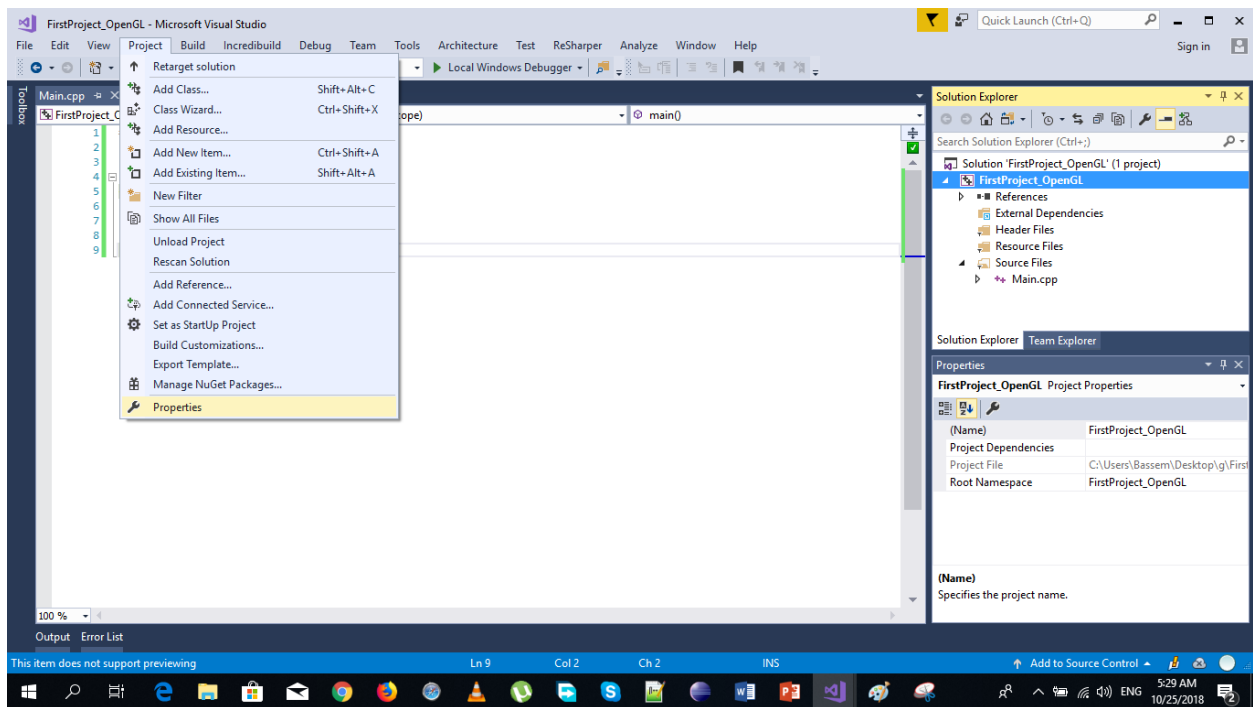
ثم نسخ المجلد الذي تم ضغطه في المرحلة السابقة



بالعودة إلى فيجوال ستوديو

من خصائص المشروع من القائمة Project-> Properties

أو من الضغط بالزر الأيمن على المشروع (FirstProject\_OpenGL) << Properties



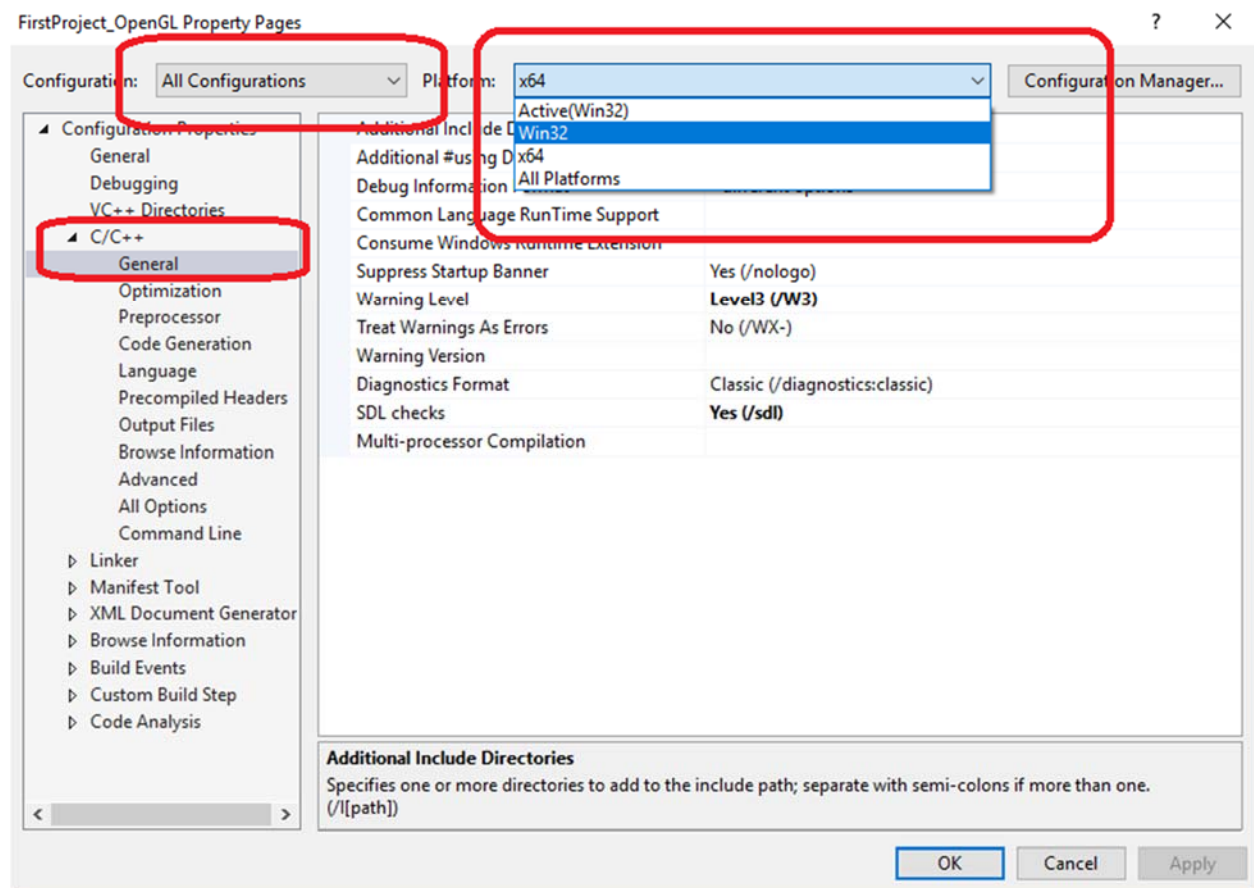


من اللوحة اليمنى القائمة

C/C++ → General

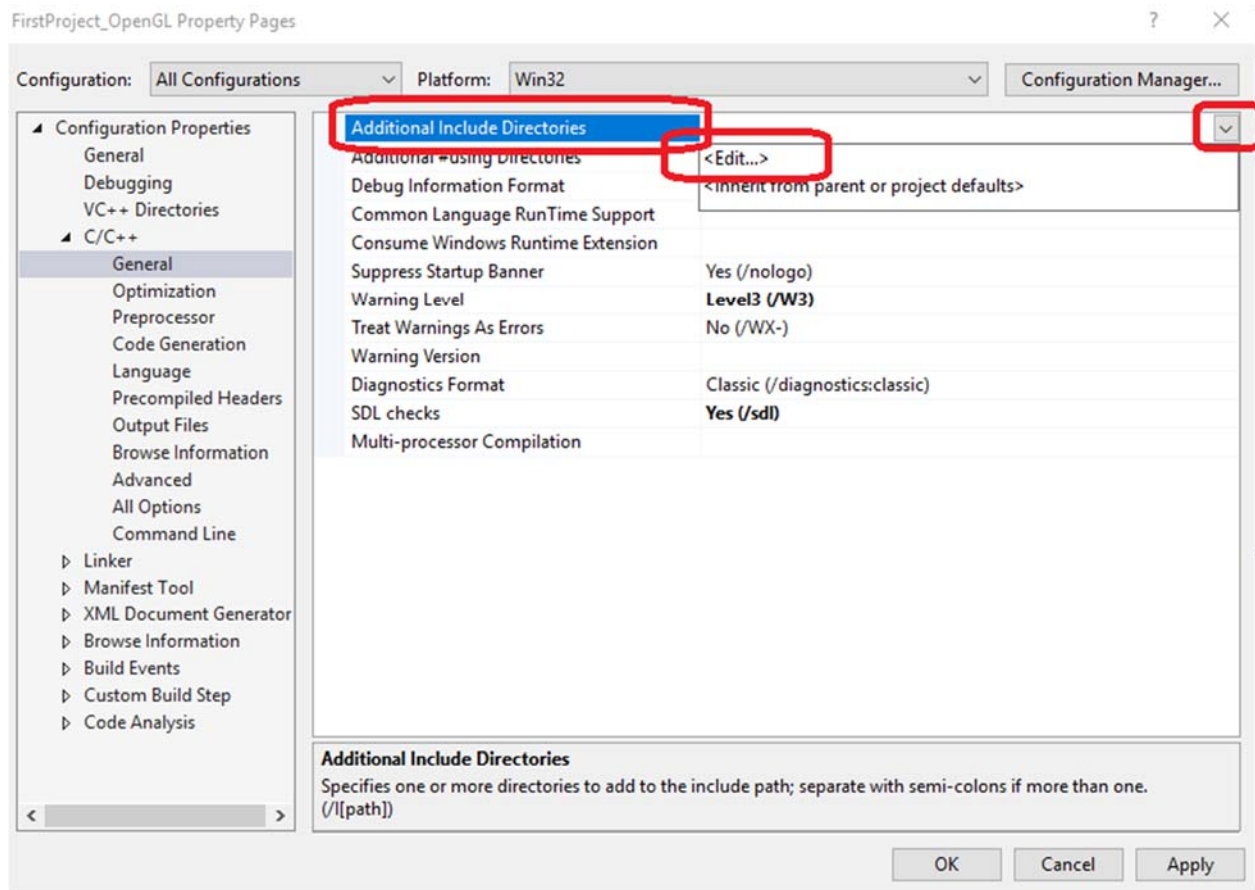
التأكد من أن Configuration تحتوي على All Configuration

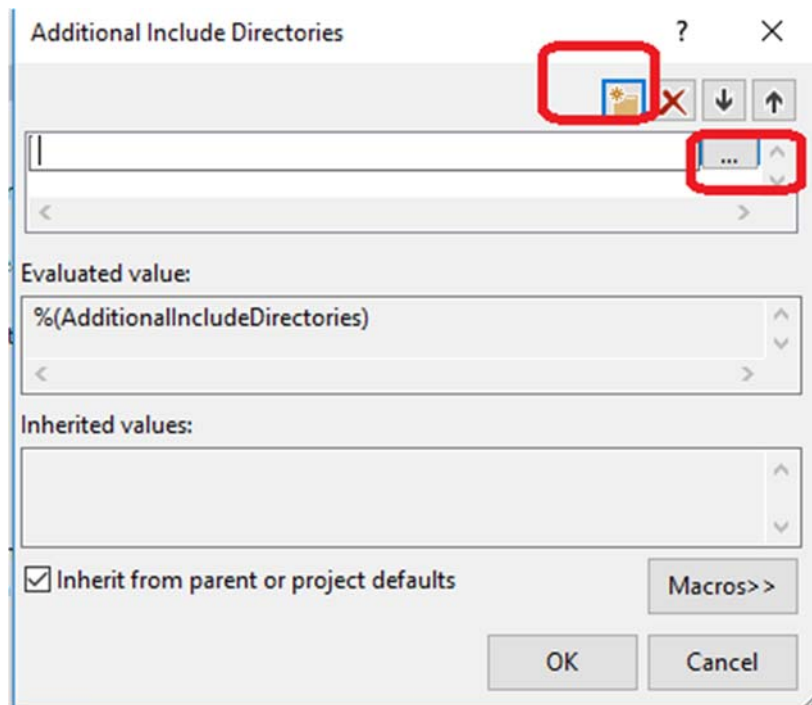
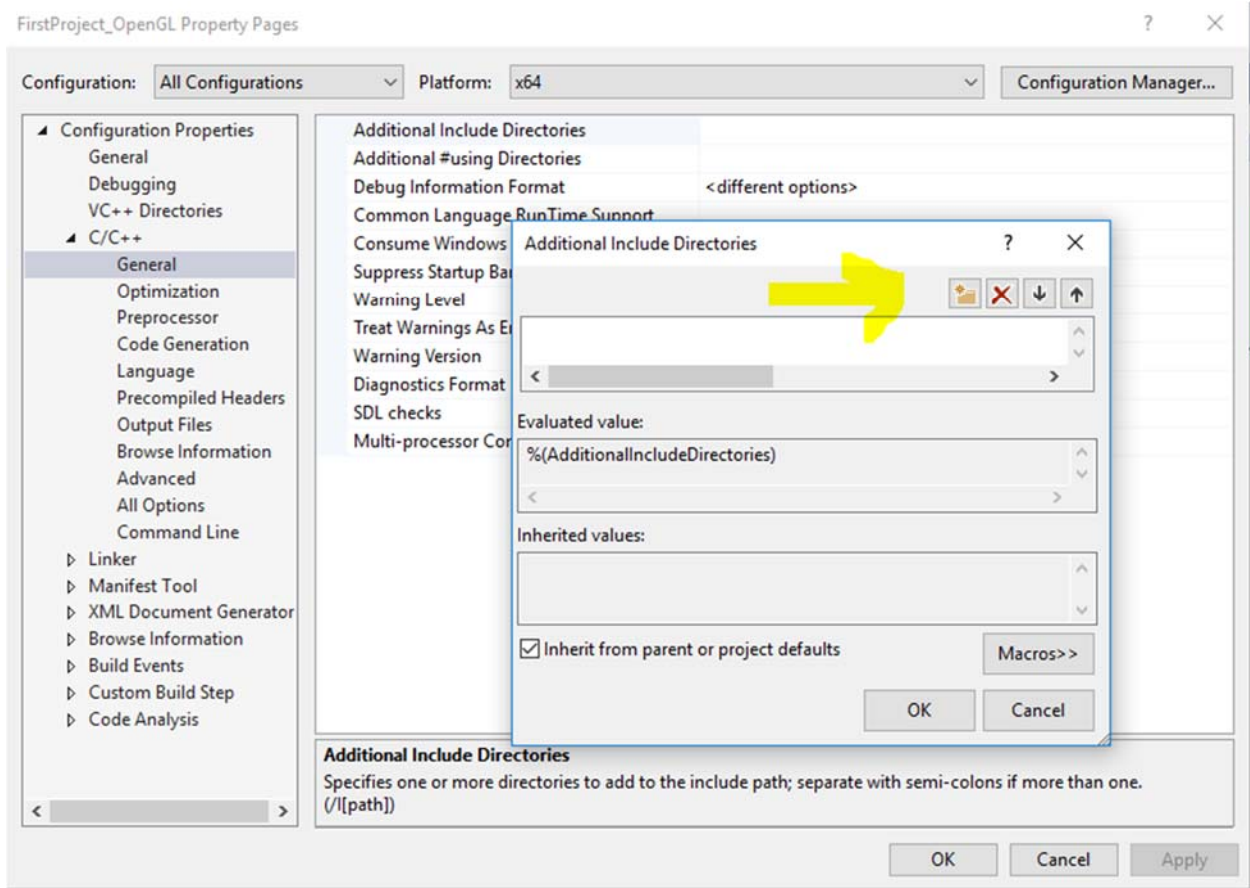
و Platform تحتوي على نوع المعالج المراد العمل عليه، غالباً Win32



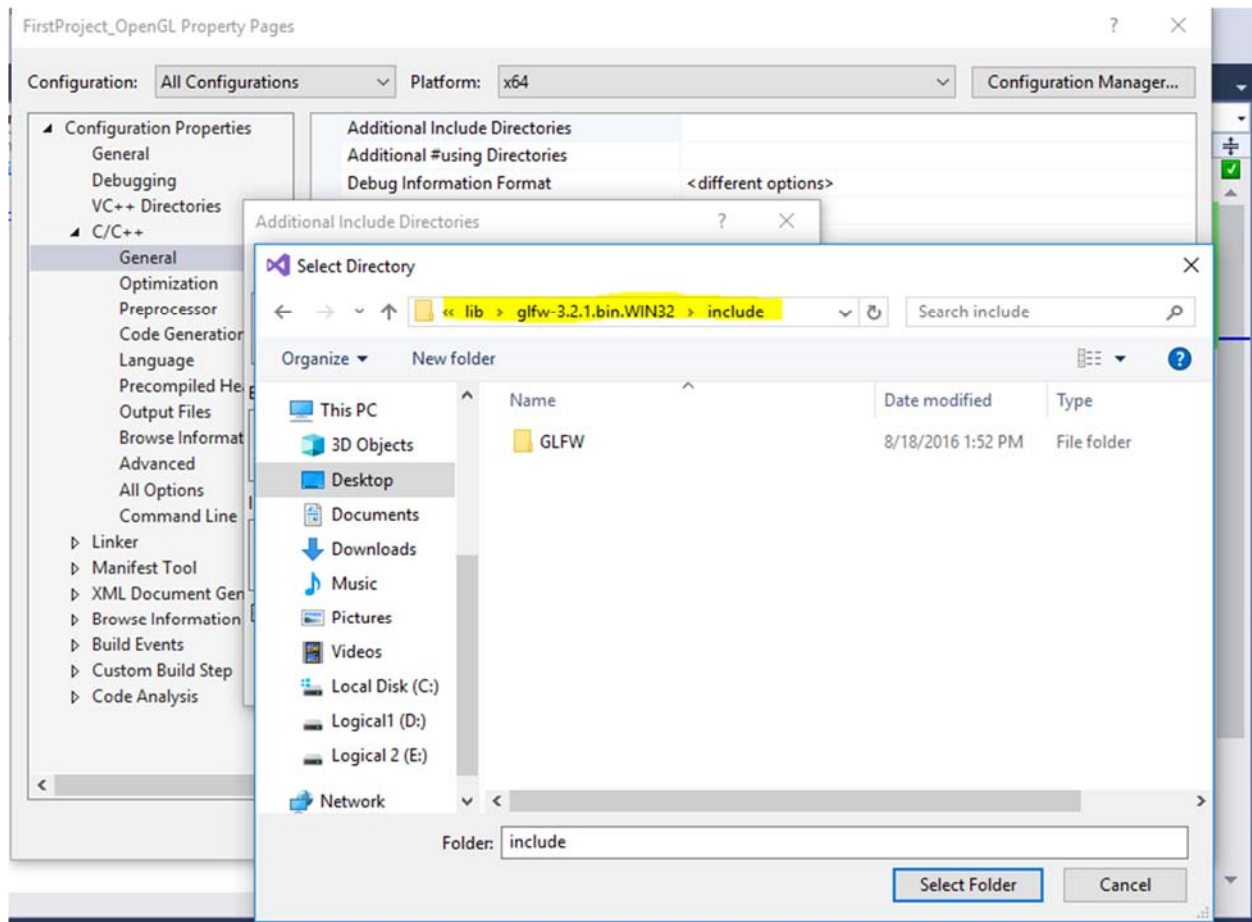
## اختيار Additional Include Directories

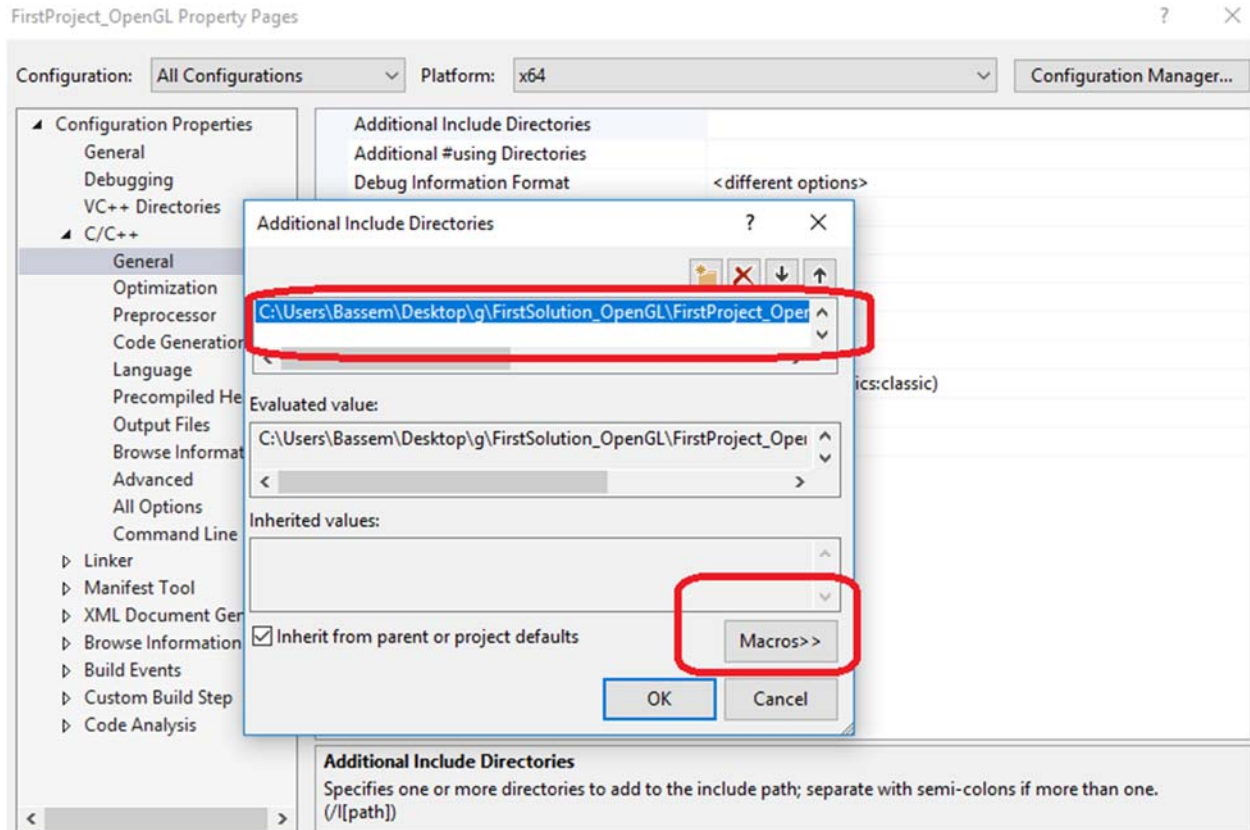
والضغط على السهم الصغير أقصى اليمين ثم Edit

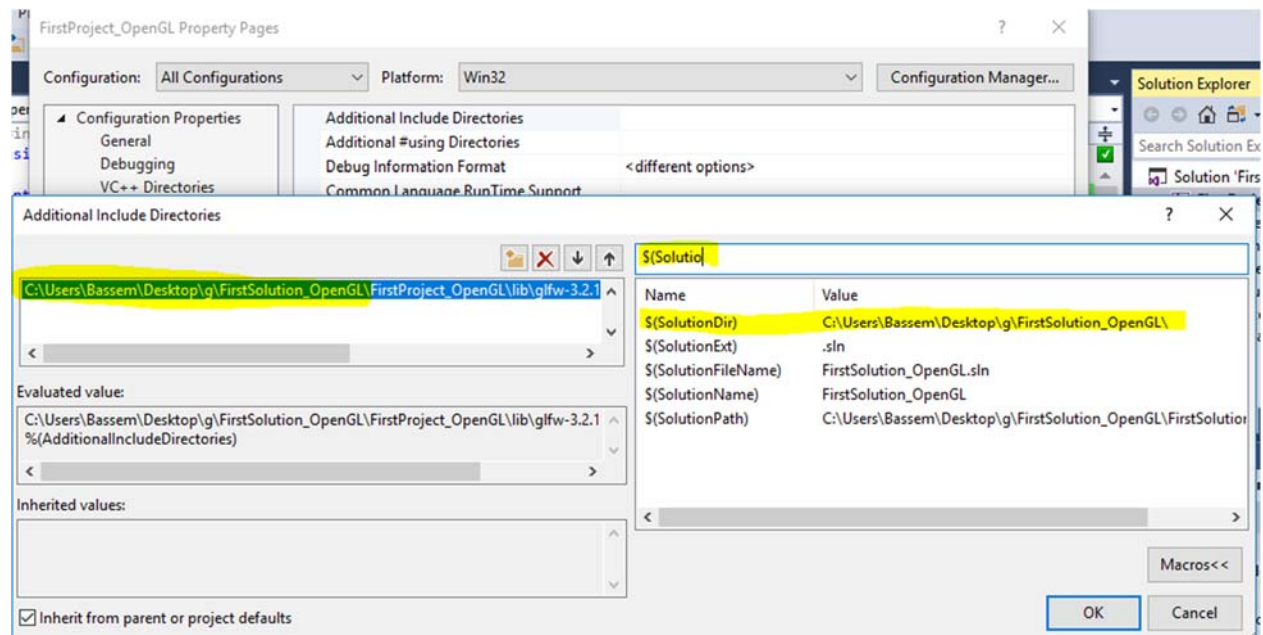
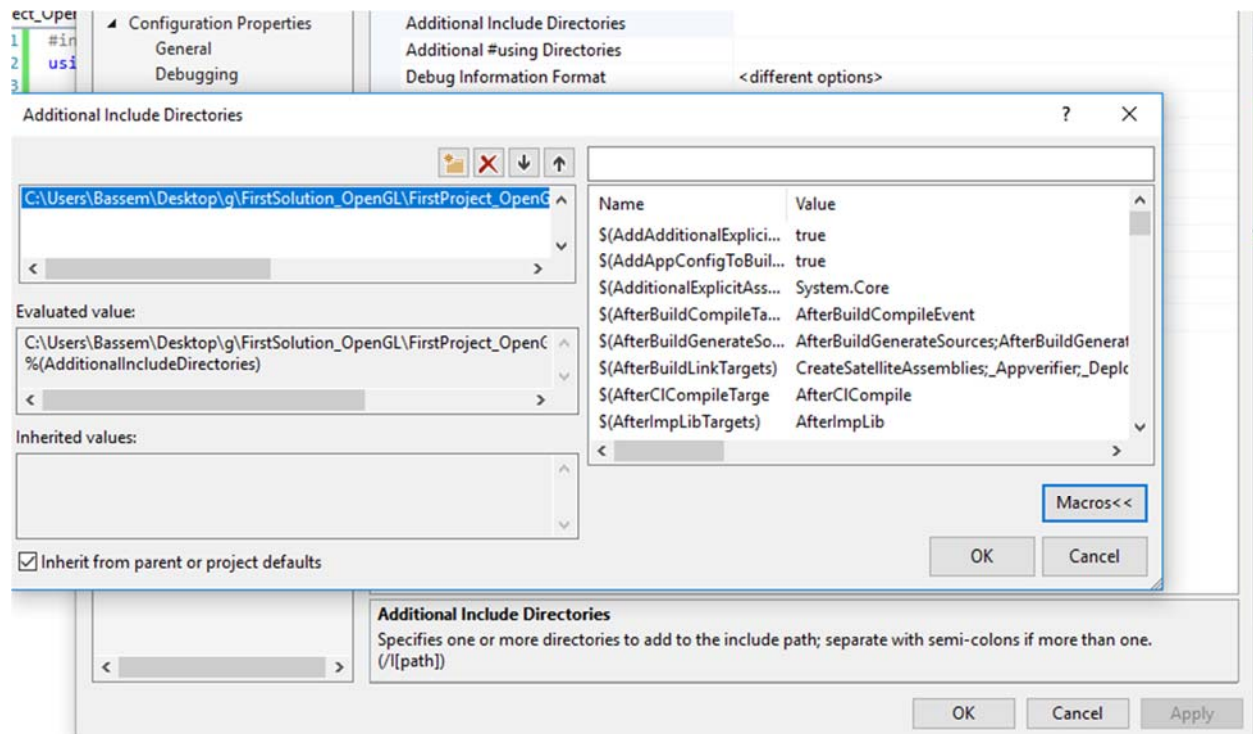




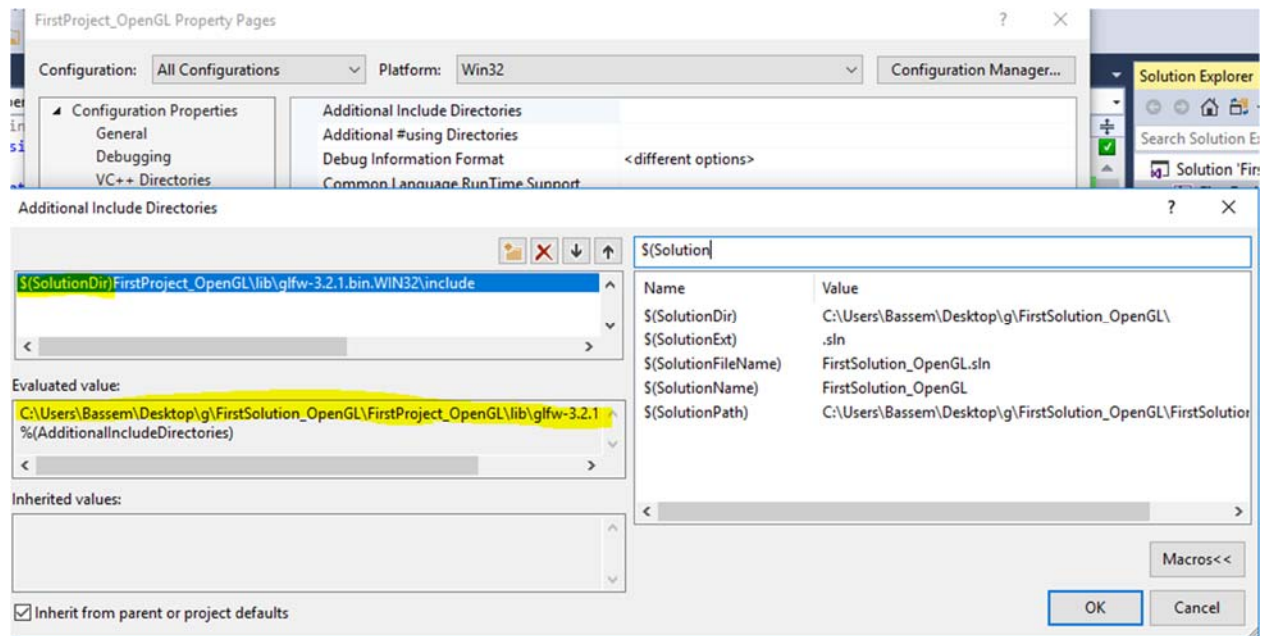
البحث عن المسار FirstSolution\_OpenGL\FirstProject\_OpenGL\lib\glfw-3.2.1.bin.WIN32





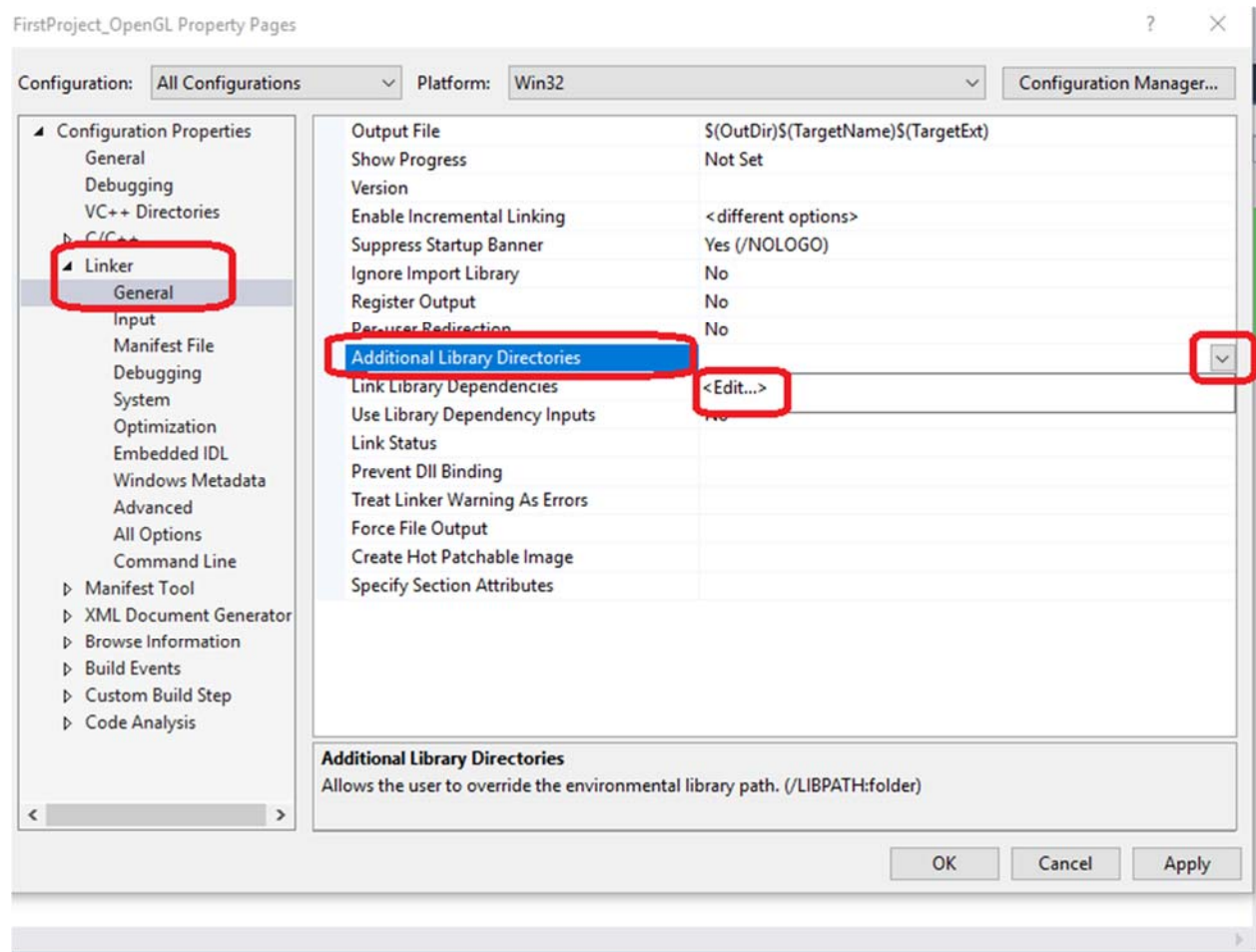






من اللوحة اليمنى الشاشة

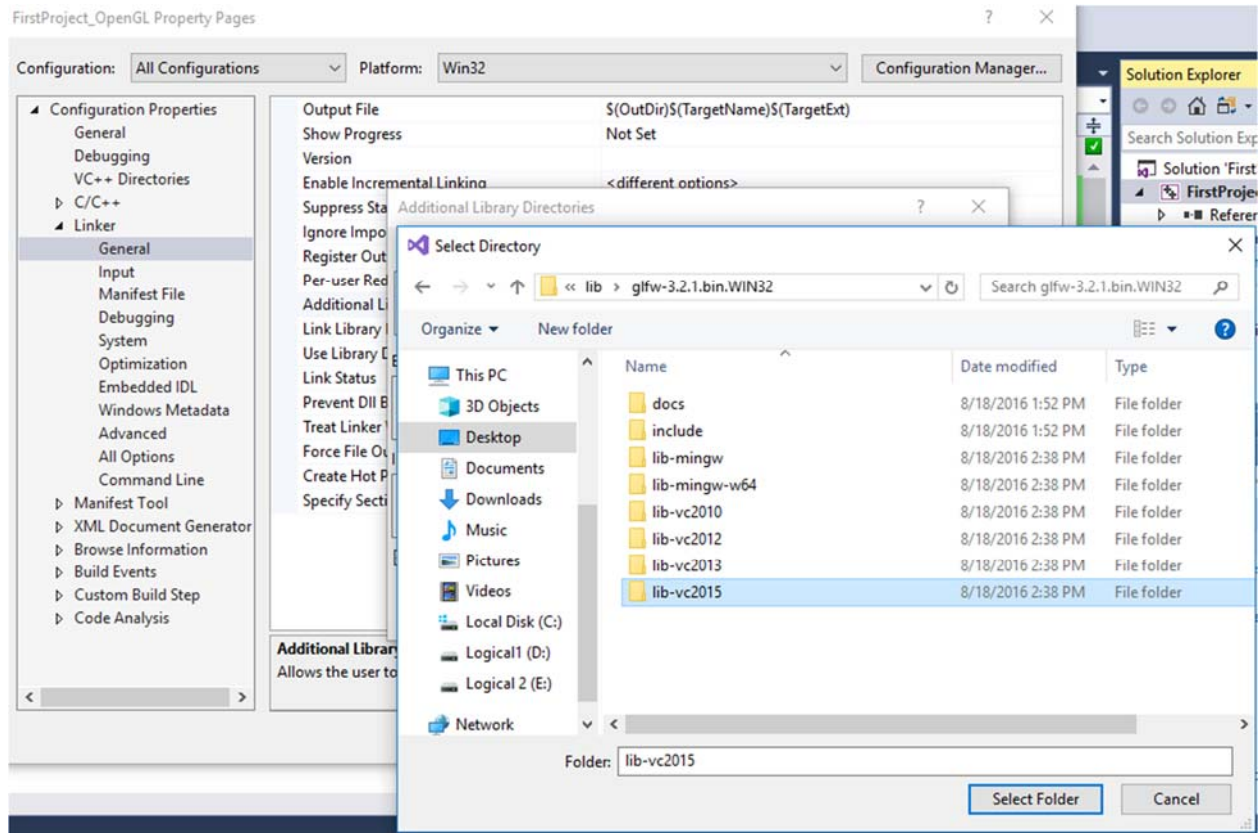
Linker → General → Additional Library Directories

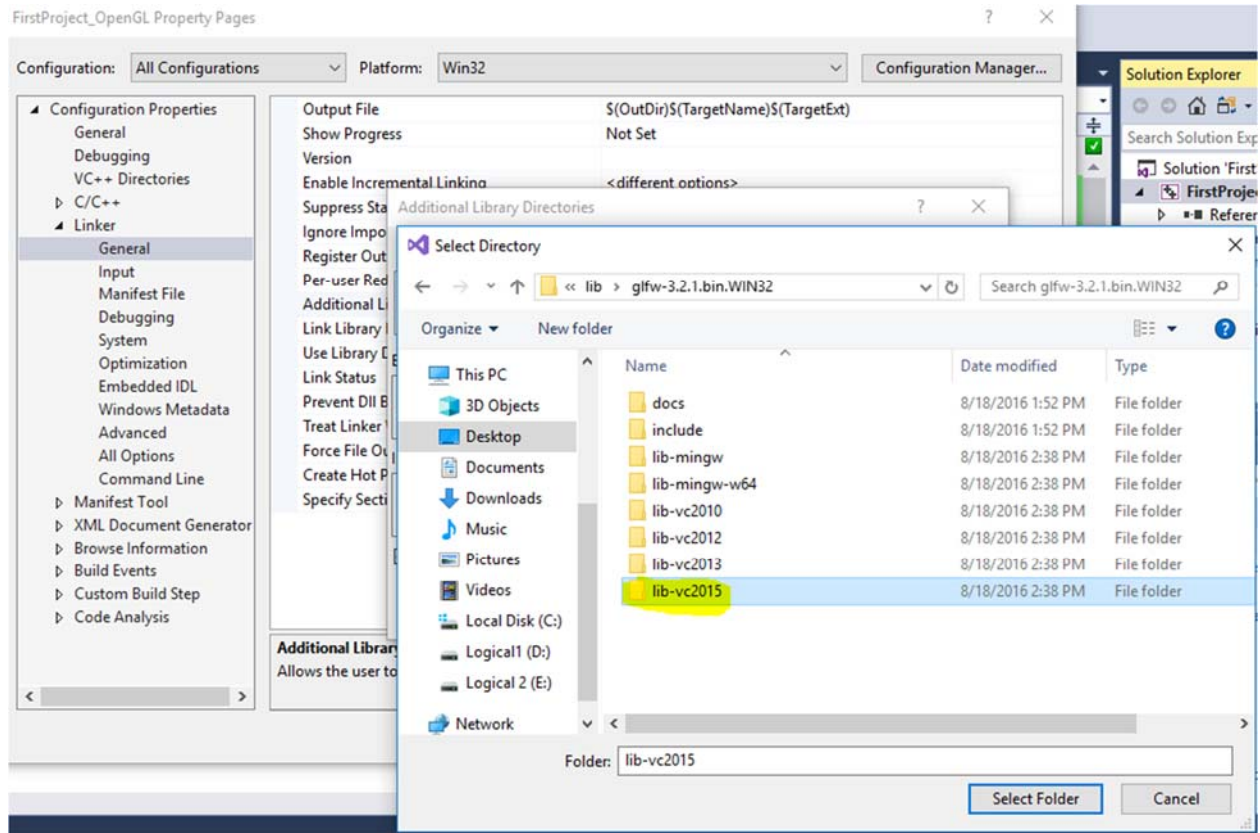


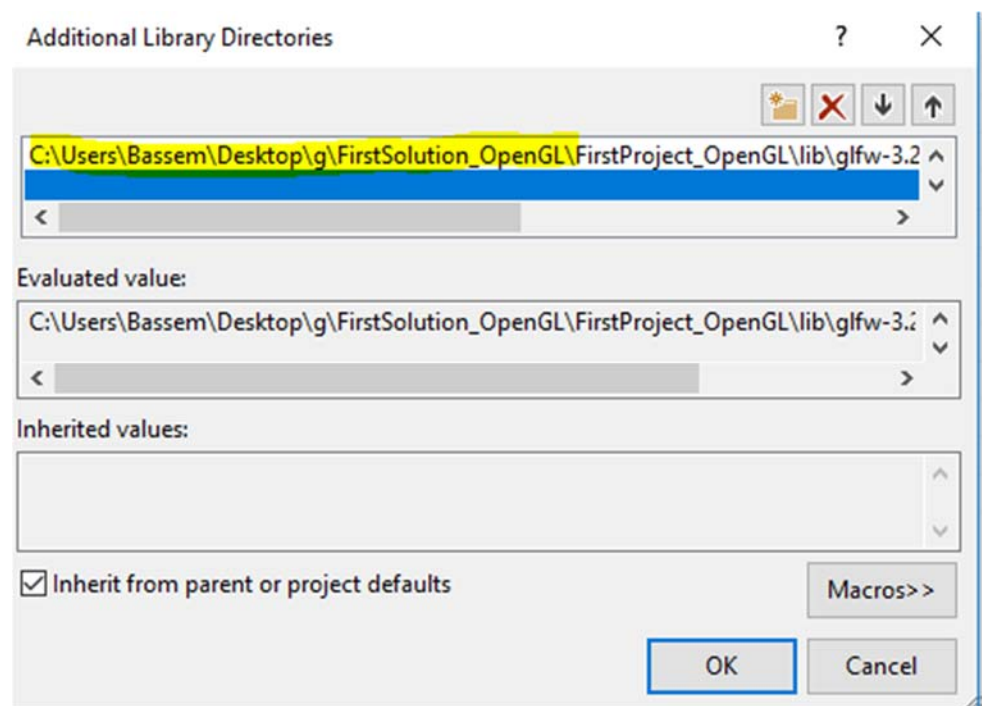
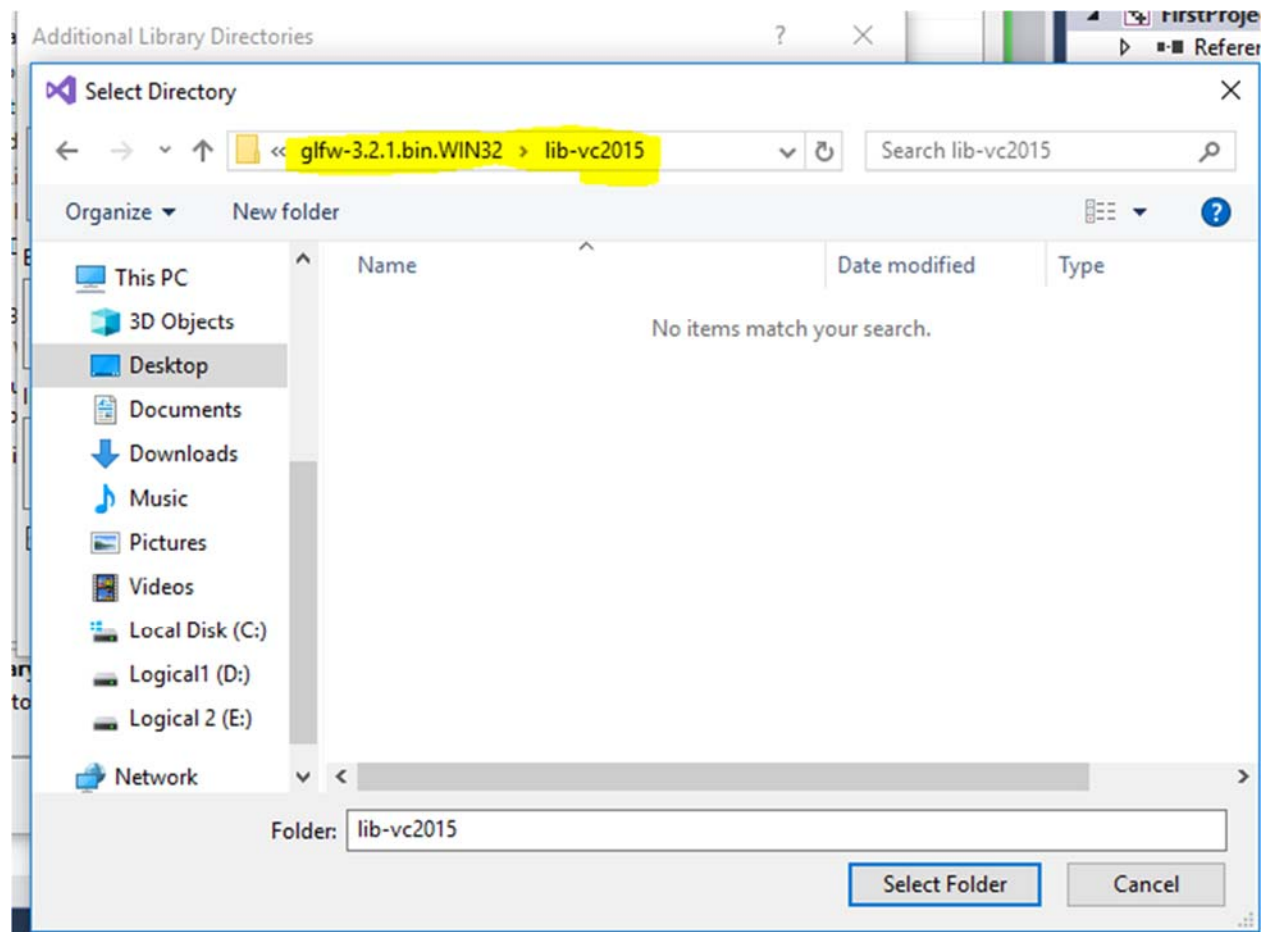


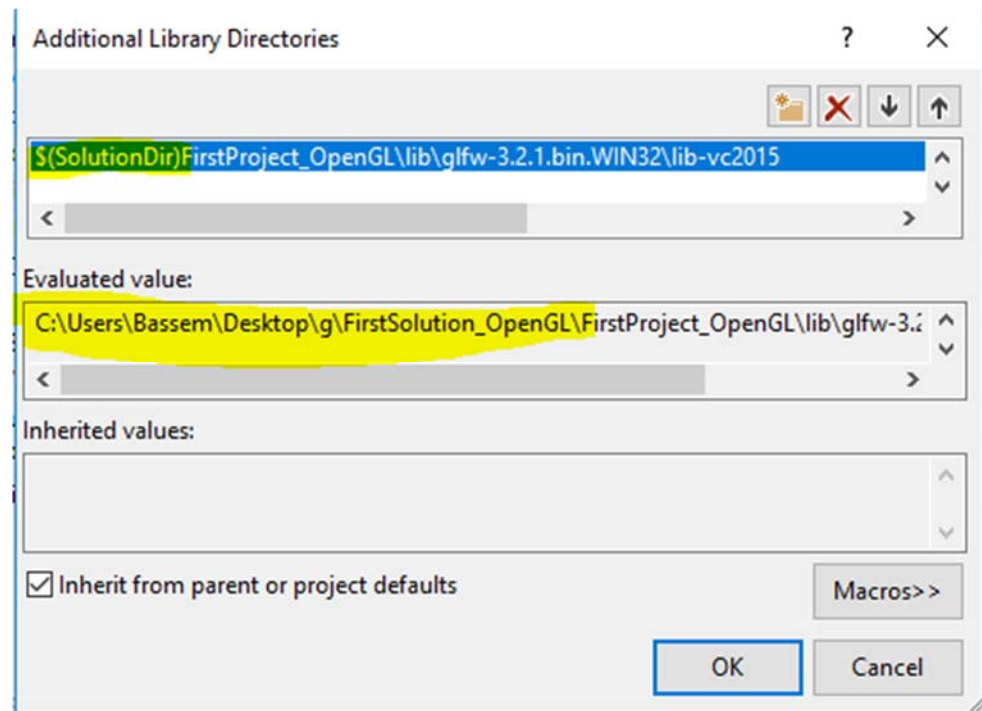
حسب إصدار فيجوال ستوديو يمكن اختيار المكتبة المناسبة من المسار  
FirstSolution\_OpenGL\FirstProject\_OpenGL\lib\glfw-3.2.1.bin.WIN32

في حالة 2015 و 2017 يمكن اختيار lib\_vc2015

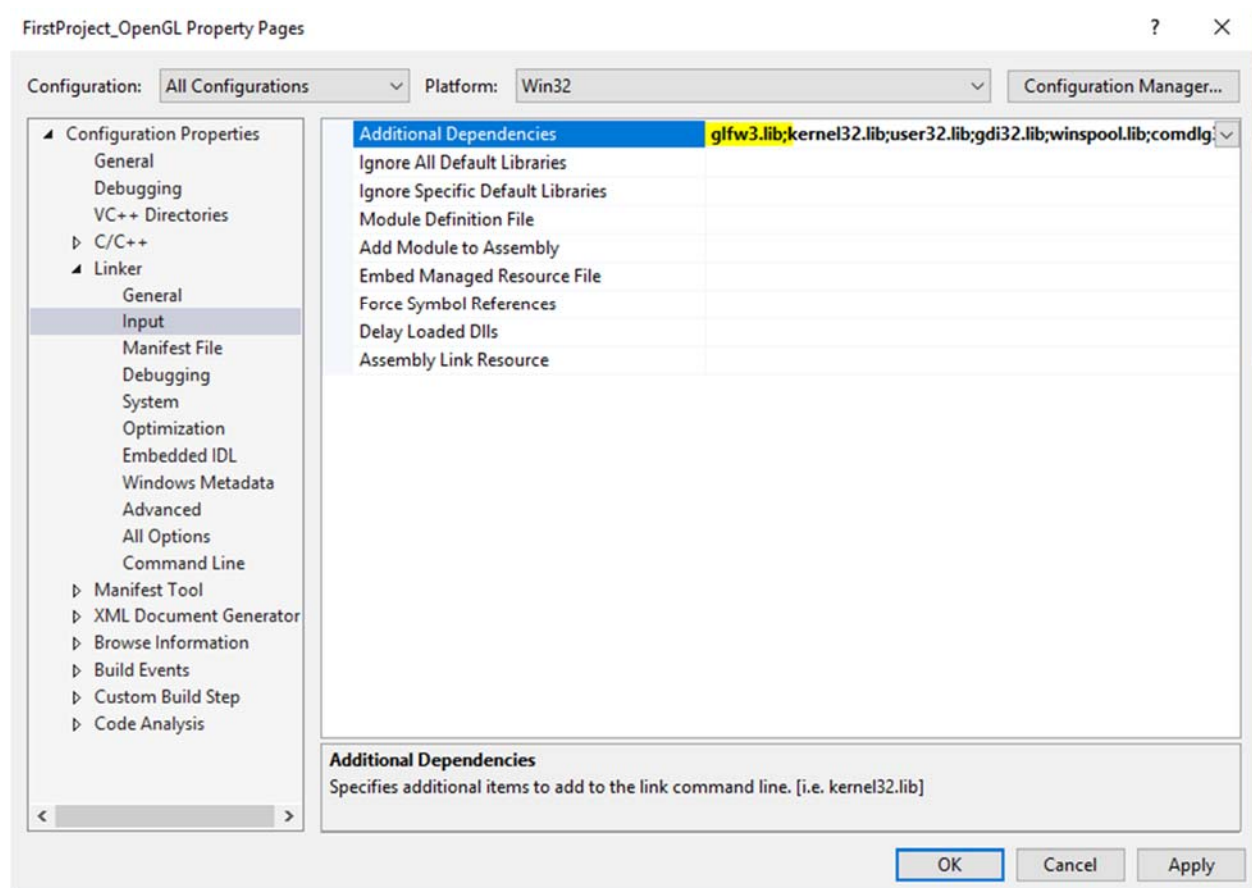


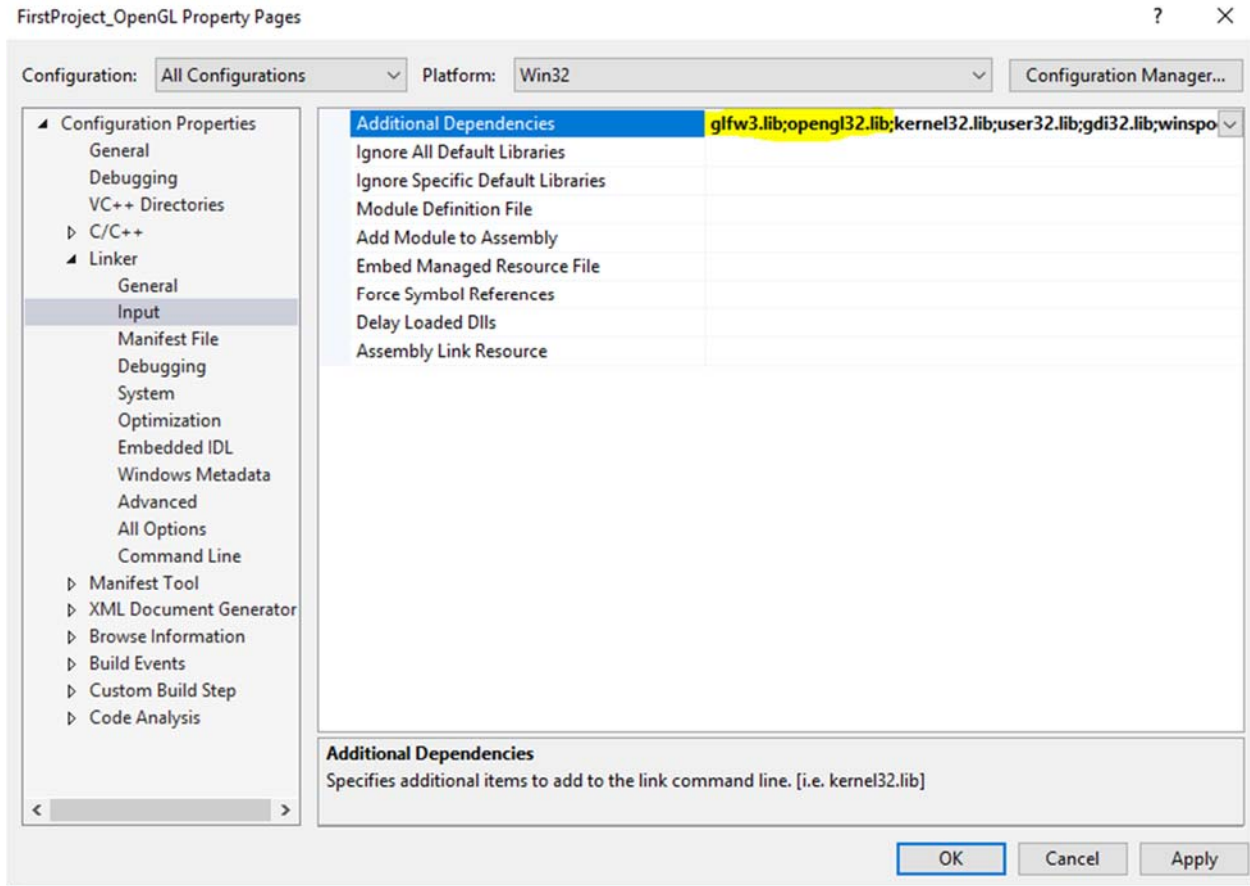




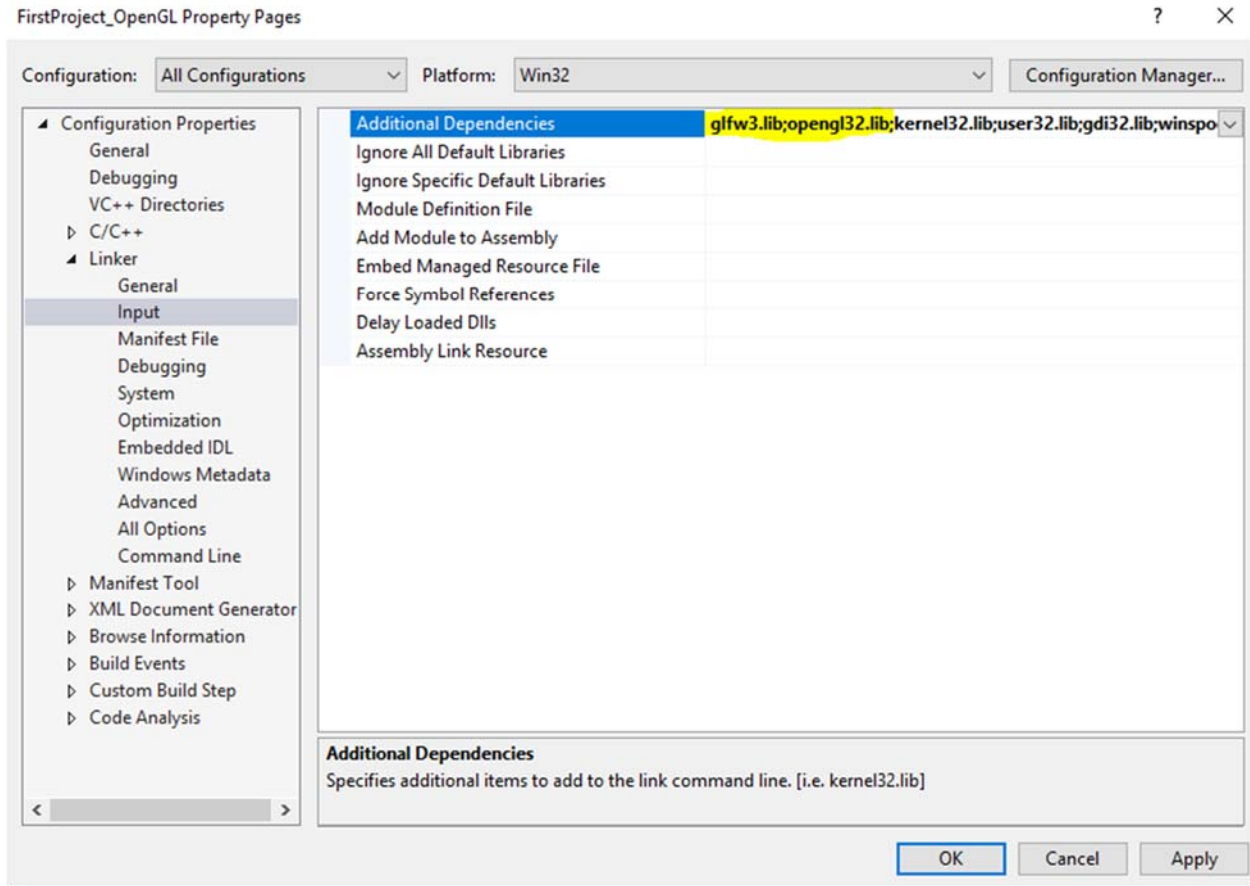














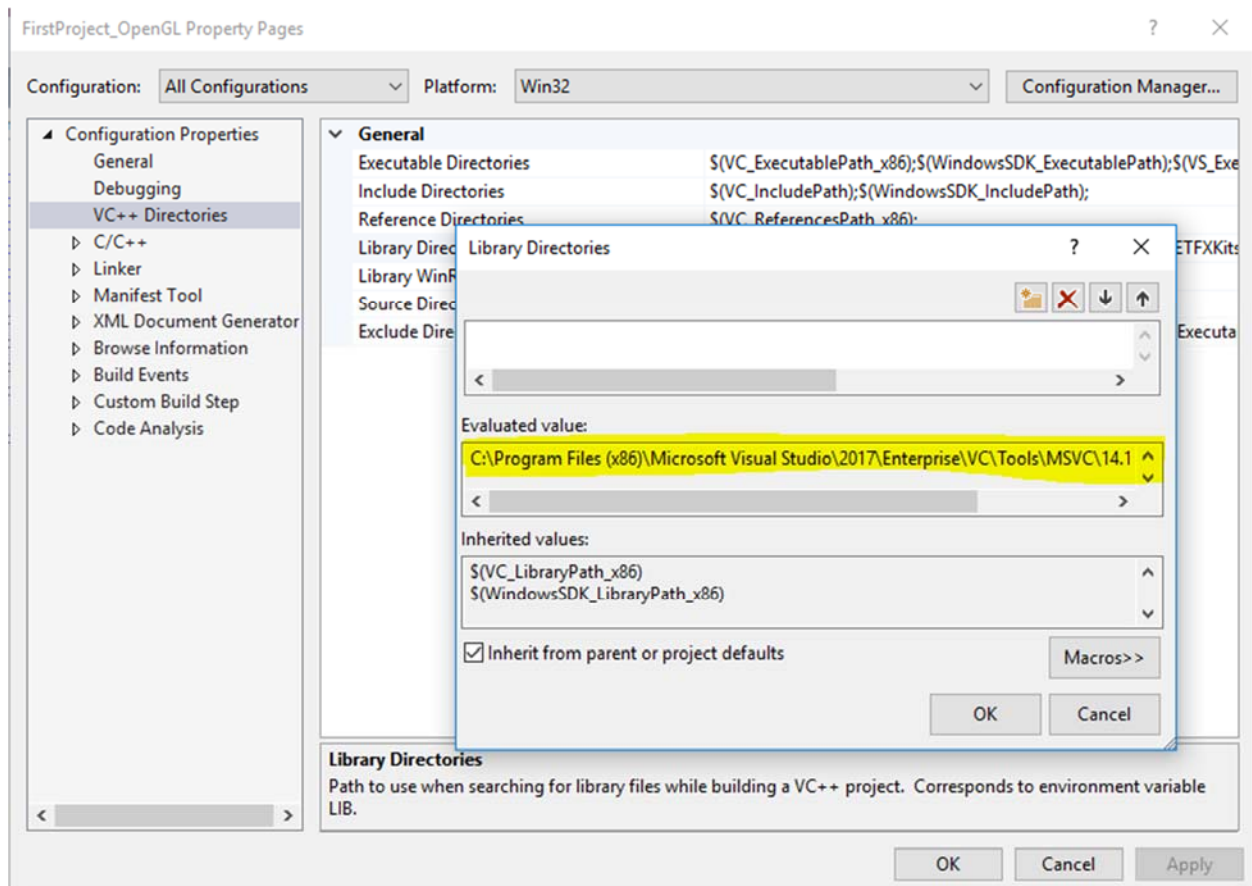
ملاحظة: في حال عدم نجاح الخطوات السابقة المحاولة لتضمين المكتبات في فيجوال ستوديو حيث يمكن البحث عن مسارات lib التي يجب نسخ ملفات .lib إليها حسب الخطوات التالية، والاستغناء عن مرحلة

Linker → General → Additional Library Directories

أما باقي الخطوات تبق نفسها:

C/C++ → General → Additional Include Library

Linker → Input → Additional Dependencies



في حال عدم إعداد المكتبة بشكل صحيح عندئذ يمكنك استخدام CMake لإعادة ترجمة المكتبة حيث يمكن تحميل الرماز المصدري للمكتبة من الموقع السابق، أو يمكن الحصول عليه من المجلد المزود للطلاب في المجلد الفرعي src

GLFW\src

إنشاء نافذة جديدة: نسخ المثال الموجود في الموقع ولصقه في main.cpp وتشغيل البرنامج

```
#include <GLFW/glfw3.h>

int main(void)
{
    GLFWwindow* window;

    /* Initialize the library */
    if (!glfwInit())
        return -1;

    /* Create a windowed mode window and its OpenGL context */
    window = glfwCreateWindow(640, 480, "Hello World", NULL, NULL);
    if (!window)
    {
        glfwTerminate();
        return -1;
    }

    /* Make the window's context current */
    glfwMakeContextCurrent(window);

    /* Loop until the user closes the window */
    while (!glfwWindowShouldClose(window))
    {
        /* Render here */
```

```
glClearColor(GL_COLOR_BUFFER_BIT);

/* Swap front and back buffers */

glfwSwapBuffers(window);

/* Poll for and process events */

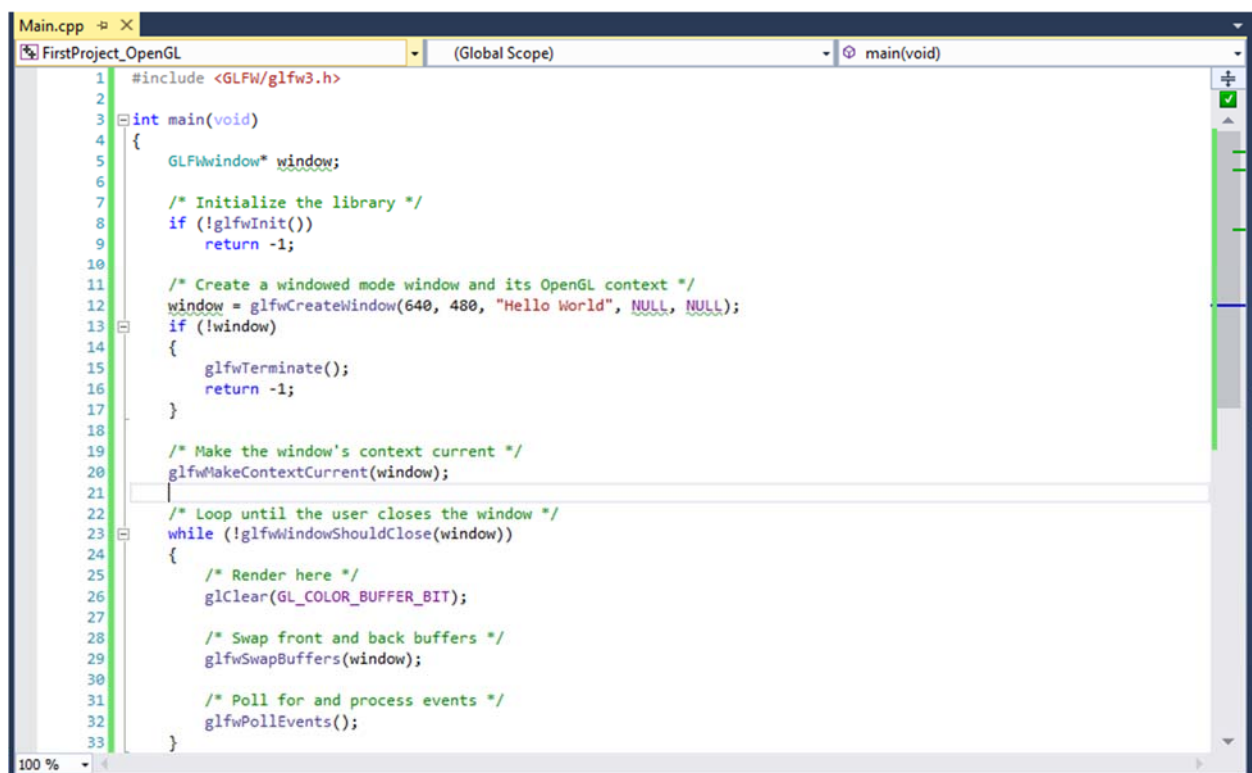
glfwPollEvents();

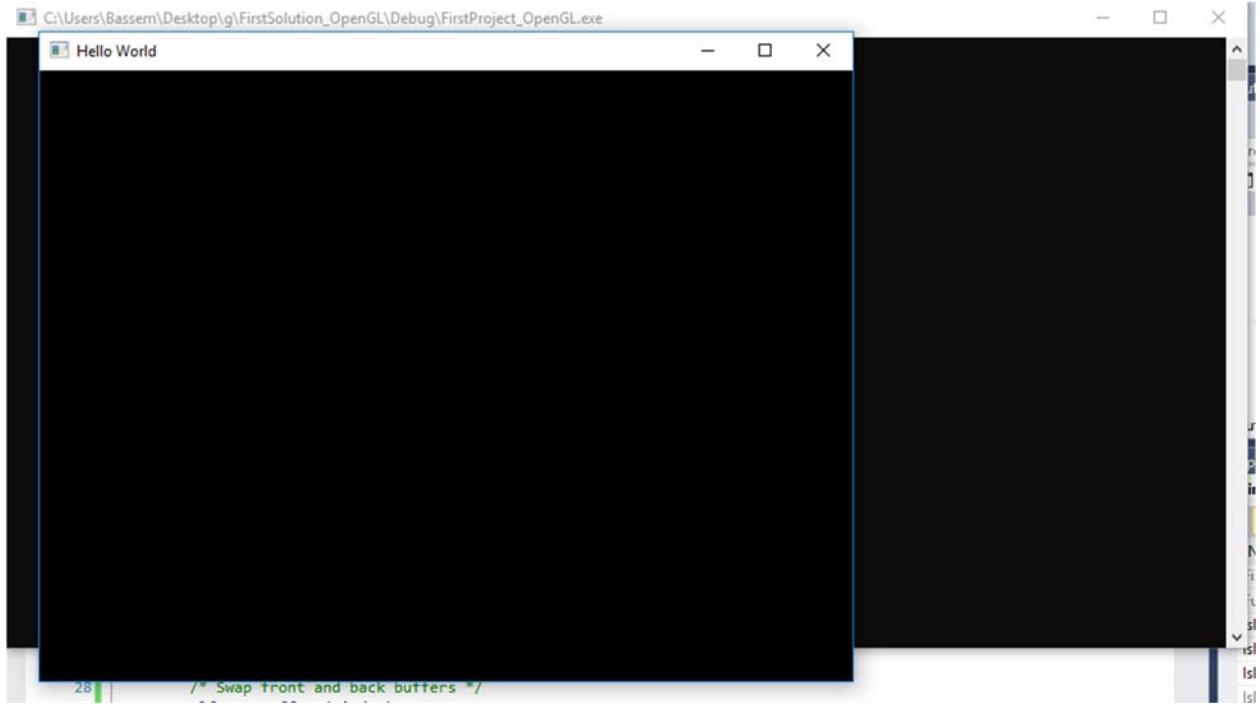
}

glfwTerminate();

return 0;

}
```





يمكن إضافة التعليمات التالية لرسم مثلث

```
if (!window)
{
    glfwTerminate();
    return -1;
}

/* Make the window's context current */
glfwMakeContextCurrent(window);

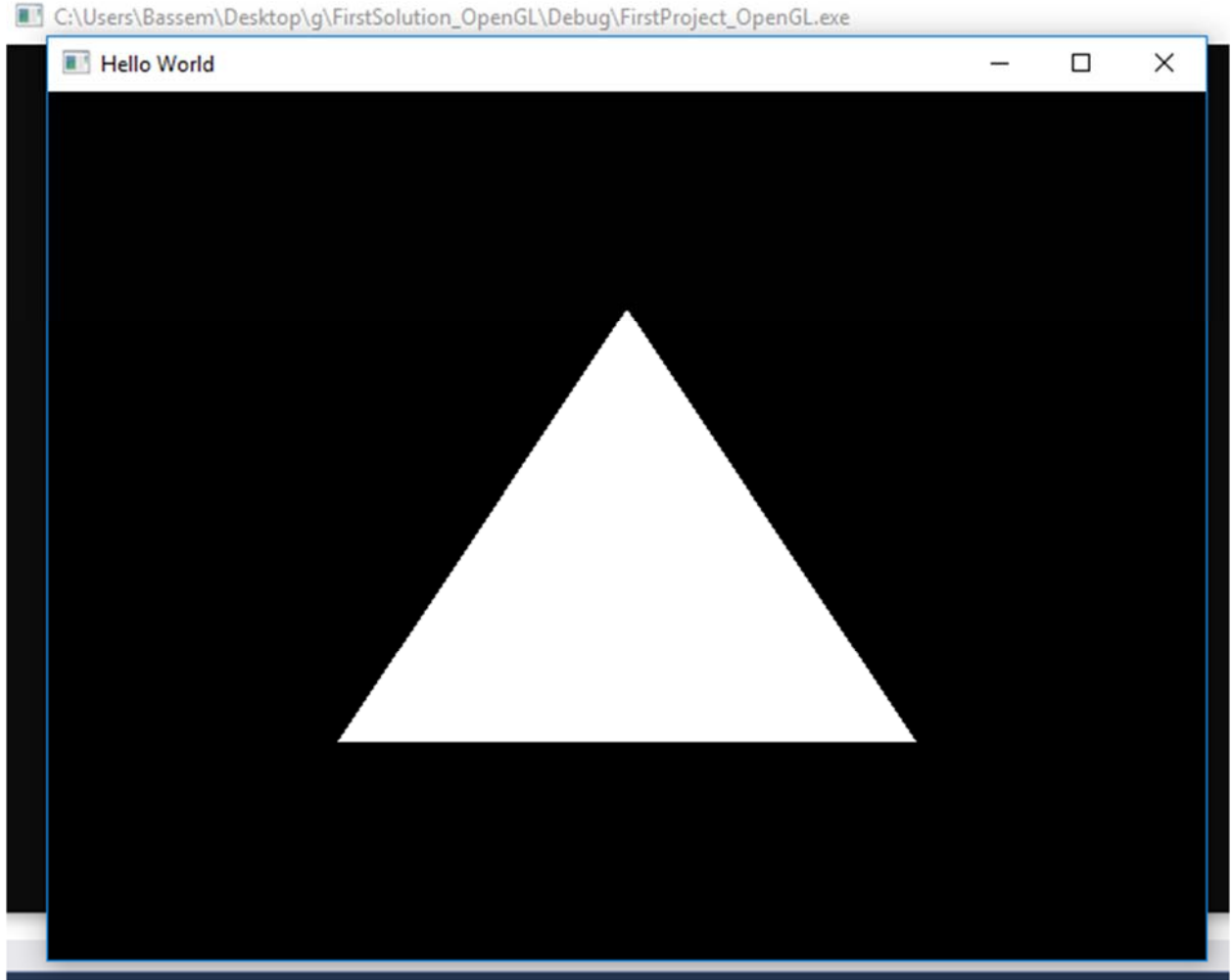
/* Loop until the user closes the window */
while (!glfwWindowShouldClose(window))
{
    /* Render here */
    glClear(GL_COLOR_BUFFER_BIT);

    glBegin(GL_TRIANGLES);
    glVertex2f(-0.5f, -0.5f);
    glVertex2f(0.0f, 0.5f);
    glVertex2f(0.5f, -0.5f);
    glEnd();

    /* Swap front and back buffers */
    glfwSwapBuffers(window);

    /* Poll for and process events */
    glfwPollEvents();
}

glfwTerminate();
return 0;
}
```



يمكن الوصول إلى توصيف التوابيع الخاصة بـ GLFW في المجلد

**FirstSolution\_OpenGL\FirstProject\_OpenGL\lib\glfw-3.2.1.bin.WIN32\docs\html**

والبحث عن الملف **index** وهنا يمكنك استخدام توابيع التعامل مع النوافذ وعمليات التفاعل مع الدخول والخروج، كتلقي حدث الضغط زر الفأرة أو زر من لوحة المفاتيح والتحكم بحجم النافذة.