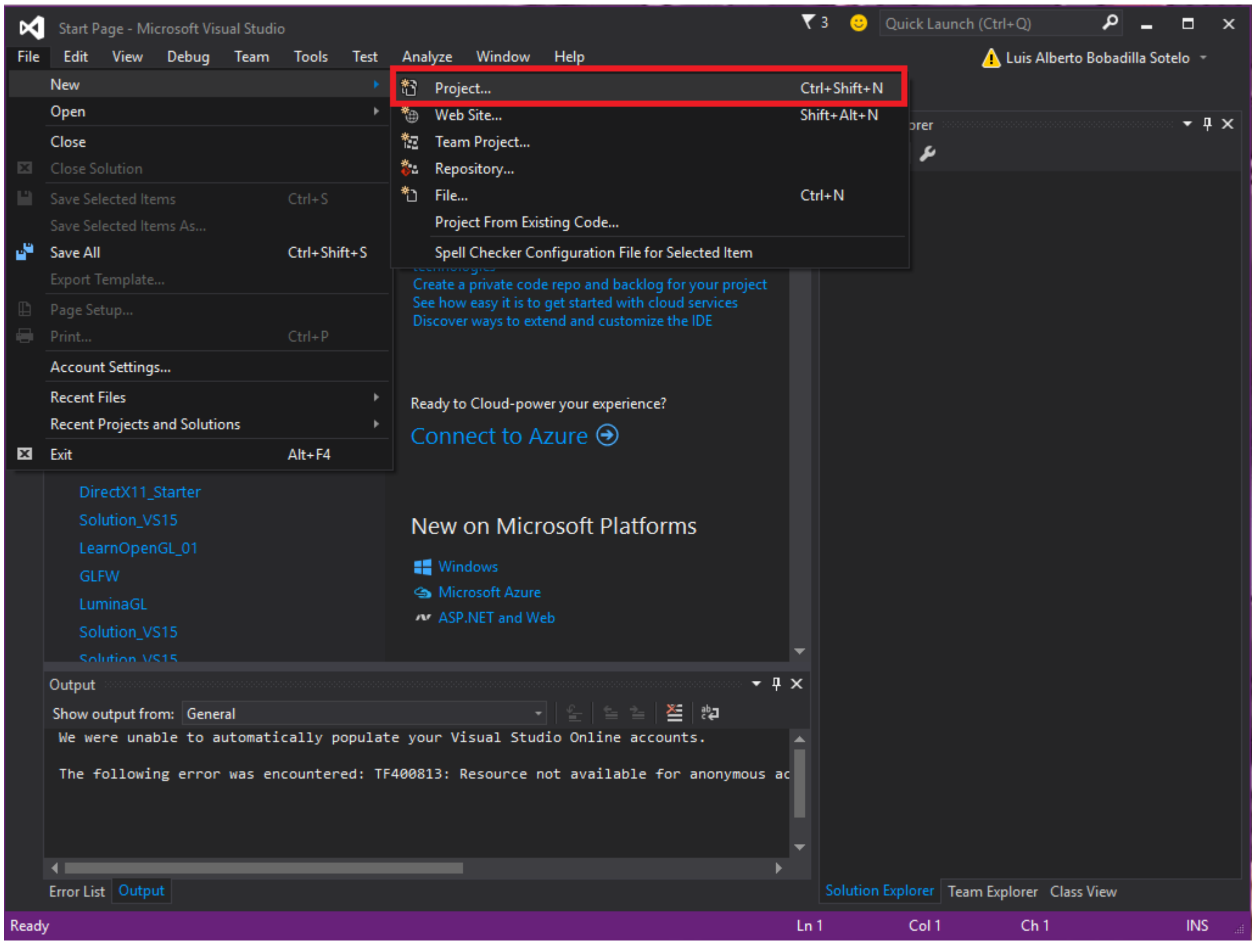
Rochester Institute of Technology
Golisano College of Computing and Information Sciences
School of Interactive Games and Media
2145 Golisano Hall – (585) 475-7680

Data Structures & Algorithms for Games & Simulation II
IGME 309
E02: Visual Studio 2017 GLFW Configuration

This exercise follows lecture 01C – OpenGL

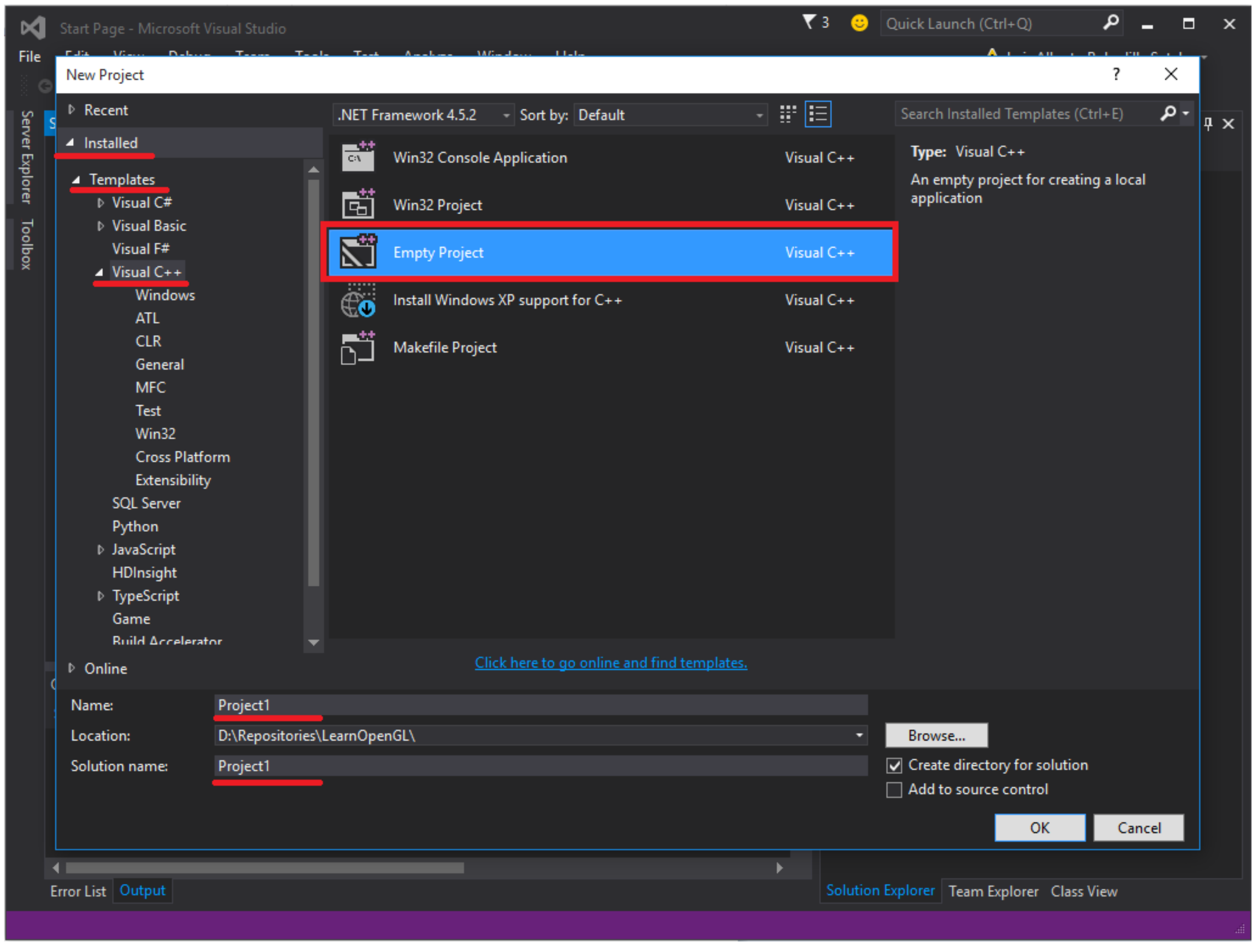
0) Instruction points 1 through 24 are the same as E01, you can just use an existing solution or start from scratch (if you do reuse a working solution please skip all the way to page 20).

1) Open a New Visual Studio 2015 Window and create a new Project (FILE/New/Project…)


Start Page - Microsoft Visual Studio
Quick Launch (Ctrl+Q)
File Edit View Debug Team Tools Test Analyze Window Help
Luis Alberto Bobadilla Sotelo
New
Open
Close
Close Solution
Save Selected Items Ctrl+S
Save Selected Items As...
Save All Ctrl+Shift+S
Export Template...
Page Setup...
Print... Ctrl+P
Account Settings...
Recent Files
Recent Projects and Solutions
Exit Alt+F4
Project... Ctrl+Shift+N
Web Site... Shift+Alt+N
Team Project...
Repository...
File... Ctrl+N
Project From Existing Code...
Spell Checker Configuration File for Selected Item
Create a private code repo and backlog for your project
See how easy it is to get started with cloud services
Discover ways to extend and customize the IDE
Ready to Cloud-power your experience?
Connect to Azure
New on Microsoft Platforms
Windows
Microsoft Azure
ASP.NET and Web
DirectX11_Starter
Solution_VS15
LearnOpenGL_01
GLFW
LuminaGL
Solution_VS15
Output
Show output from: General
Error List Output
Solution Explorer Team Explorer Class View
Ready Ln 1 Col 1 Ch 1 INS

```
We were unable to automatically populate your Visual Studio Online accounts.

The following error was encountered: TF400813: Resource not available for anonymous ac
```

2) Create a new C++ Empty Project (Installed/Visual C++/ Empty Project), give it a name (like Solution) and a location and hit OK



Type: Visual C++

An empty project for creating a local application

Name: Project1

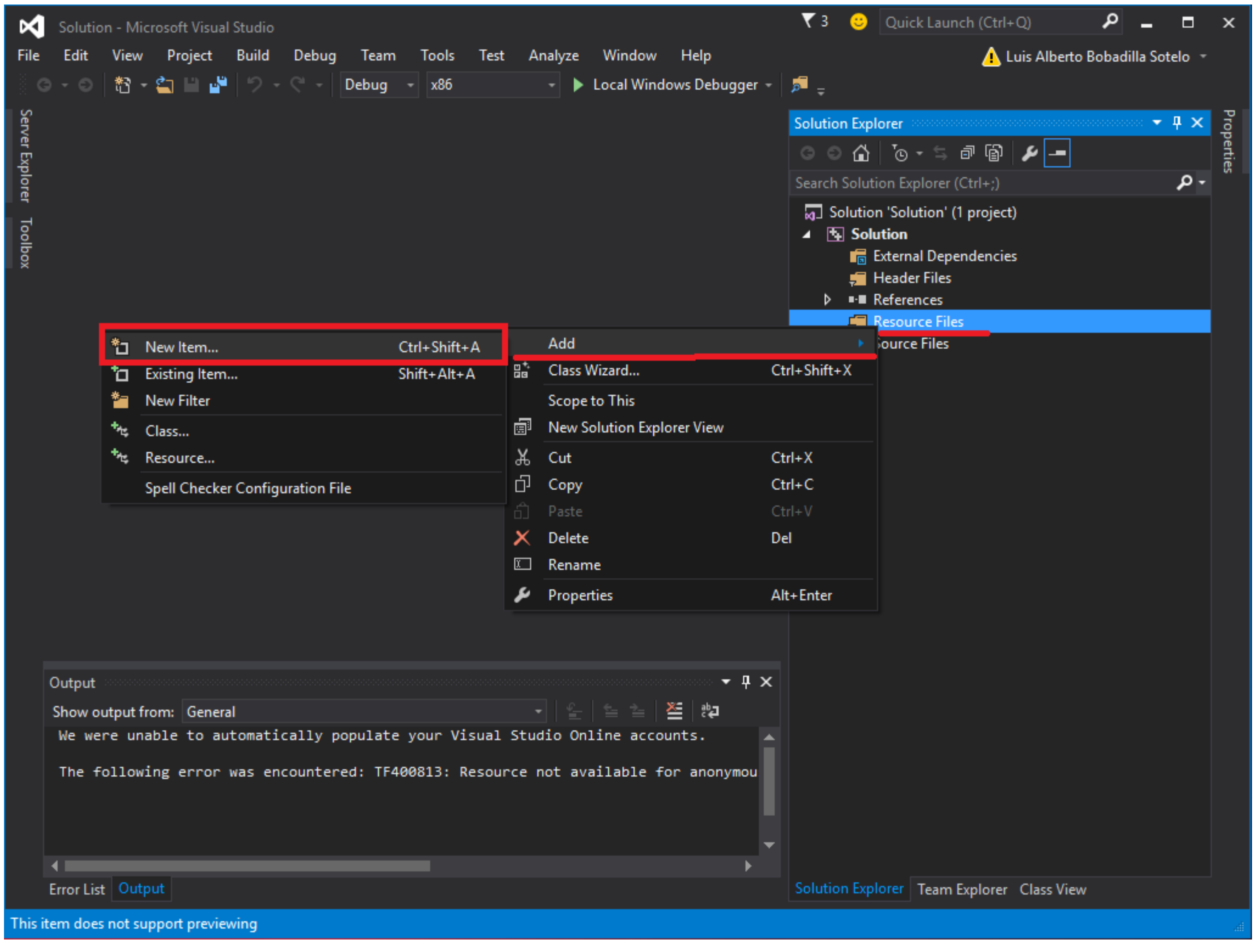
Location: D:\Repositories\LearnOpenGL\

Solution name: Project1

☑ Create directory for solution

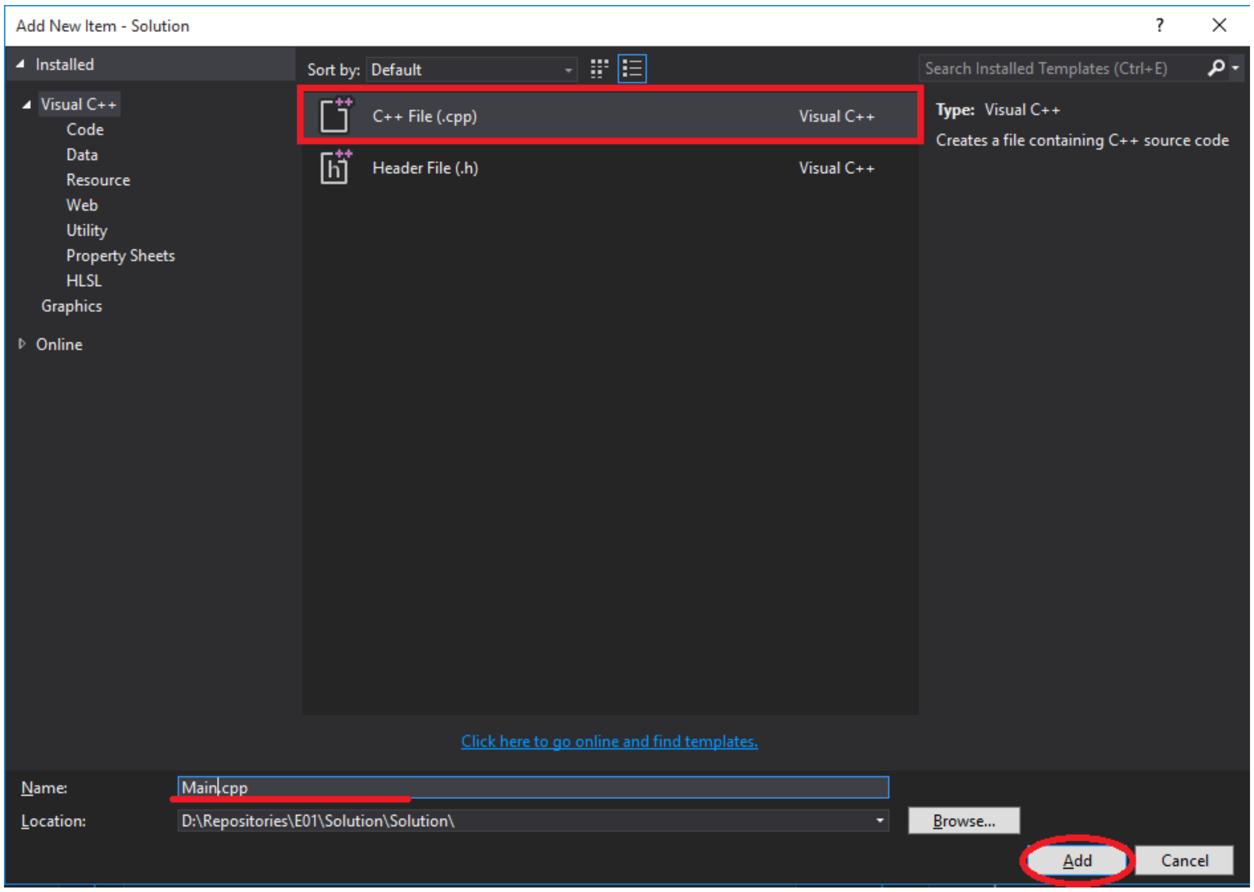
☐ Add to source control

3) Add a new C++ file IN THE SOURCE FOLDER (Source Files/Add/New Item…)


Solution - Microsoft Visual Studio
Quick Launch (Ctrl+Q)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Luis Alberto Bobadilla Sotelo
Debug x86 Local Windows Debugger
Server Explorer Toolbox
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Solution' (1 project)
Solution
External Dependencies
Header Files
References
Resource Files
Source Files
New Item... Ctrl+Shift+A
Existing Item... Shift+Alt+A
New Filter
Class...
Resource...
Spell Checker Configuration File
Add
Class Wizard... Ctrl+Shift+X
Scope to This
New Solution Explorer View
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete Del
Rename
Properties Alt+Enter
Output
Show output from: General
Error List Output
Solution Explorer Team Explorer Class View
Properties
This item does not support previewing

```
We were unable to automatically populate your Visual Studio Online accounts.

The following error was encountered: TF400813: Resource not available for anonymou
```

4) Name the file however you want (“Main” for the sake of this example) and hit OK

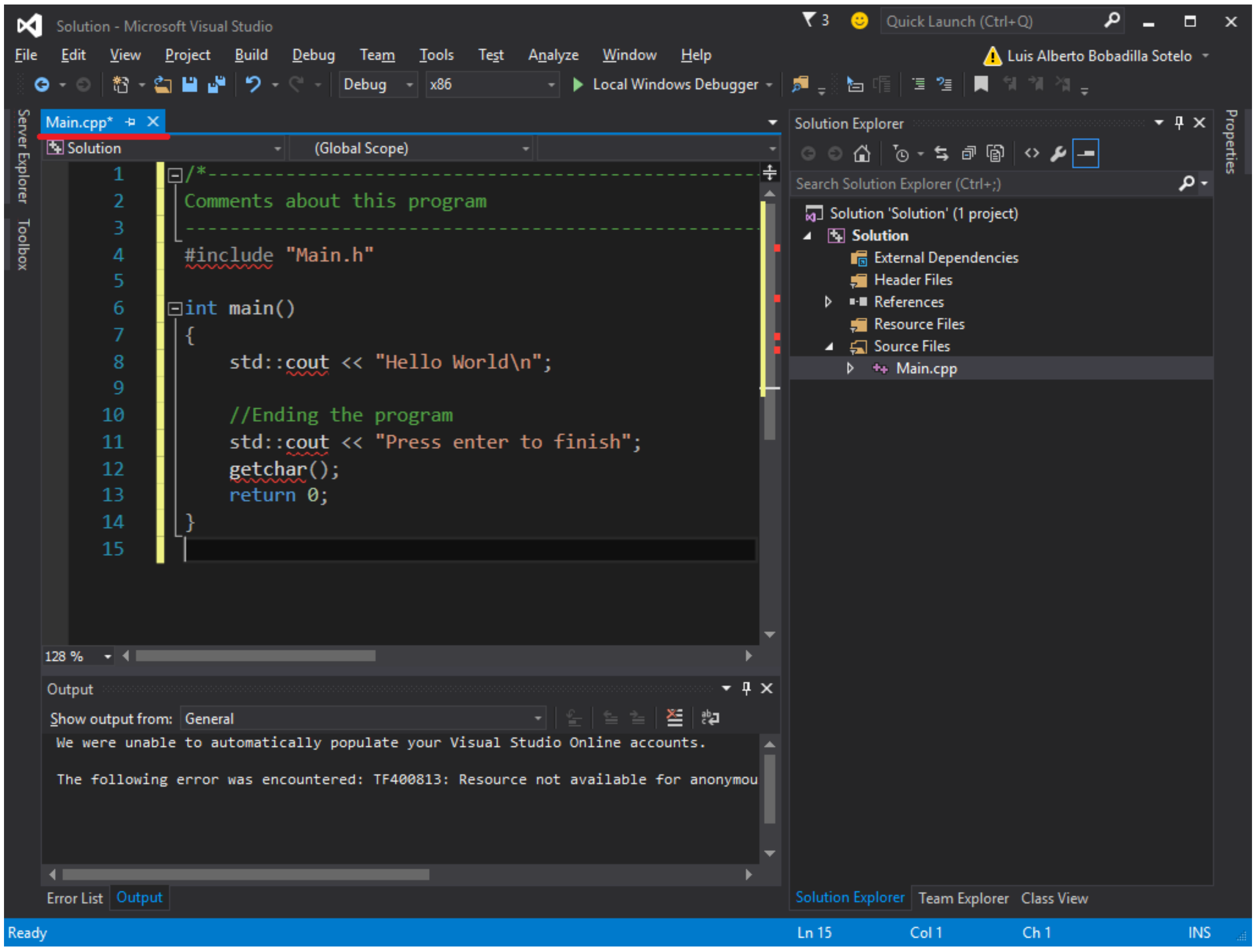

Add New Item - Solution
Installed
Visual C++
Code
Data
Resource
Web
Utility
Property Sheets
HLSL
Graphics
Online
Sort by: Default
Search Installed Templates (Ctrl+E)
C++ File (.cpp) Visual C++
Header File (.h) Visual C++
Type: Visual C++
Creates a file containing C++ source code
Click here to go online and find templates.
Browse...
Add
Cancel

Name: Main.cpp

Location: D:\Repositories\E01\Solution\Solution\

5) In Main.cpp add your main function and call your header folder, remember that #include " " and #include < > are different, the first refers to a file that is in one of the header folders local to your project and the second refers to the headers folder in Visual Studio installation directory (for this example just copy paste these lines)

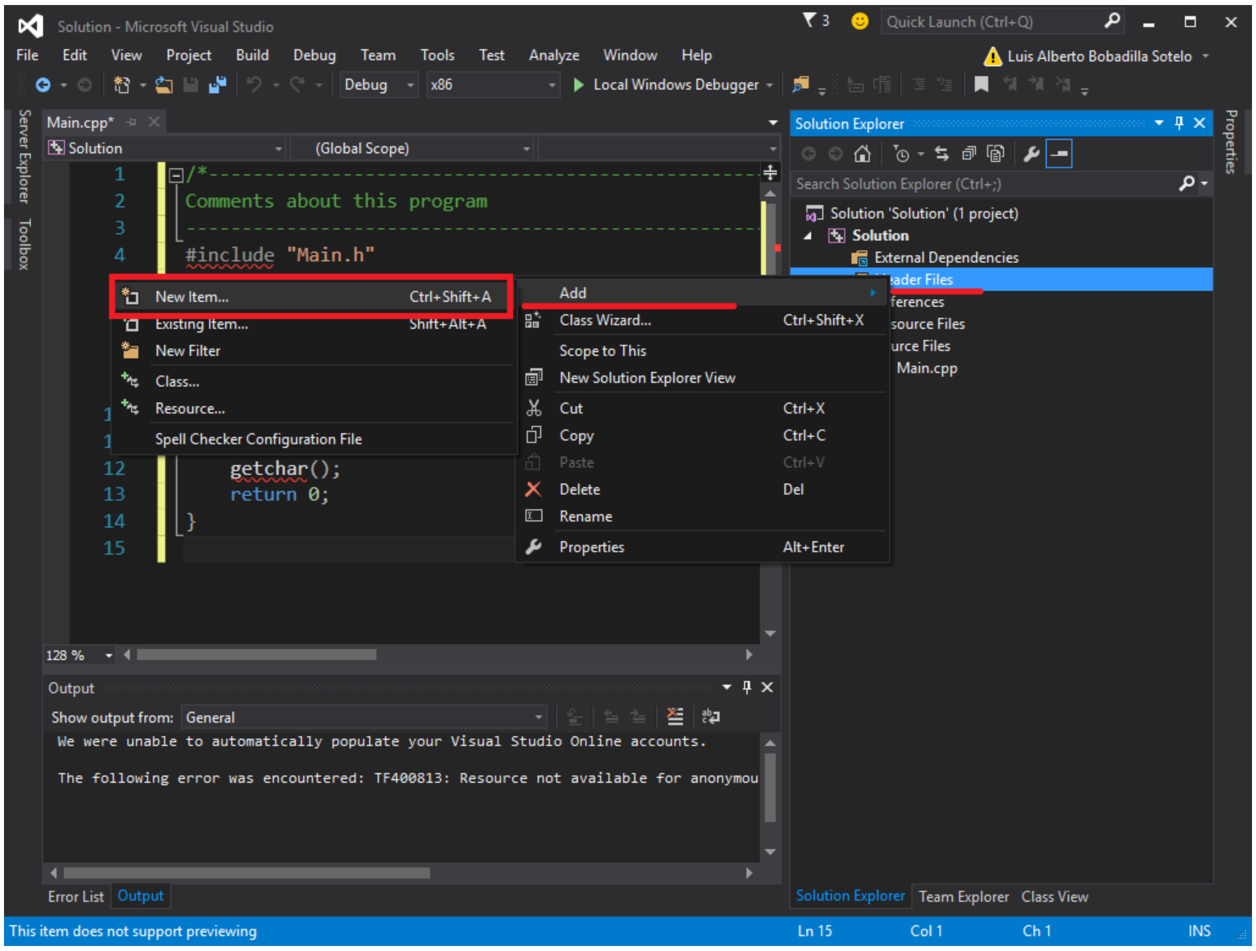
```
/*--------------------------------------------------------------------------------
-----------
Comments about this program
--------------------------------------------------------------------------------
---------*/
#include "Main.h"

int main()
{
	std::cout << "Hello World\n";

	//Ending the program
	std::cout << "Press enter to finish";
	getchar();
	return 0;
}
```



```
/*-----------------------------------------------
Comments about this program
-----------------------------------------------
#include "Main.h"

int main()
{
    std::cout << "Hello World\n";

    //Ending the program
    std::cout << "Press enter to finish";
    getchar();
    return 0;
}
```

```
We were unable to automatically populate your Visual Studio Online accounts.

The following error was encountered: TF400813: Resource not available for anonymou
```

6) Add a new header file IN THE HEADERS FOLDER (Header Files/Add/New Item…)



```
/*-------------------------------------------------------
Comments about this program
-------------------------------------------------------
#include "Main.h"
```

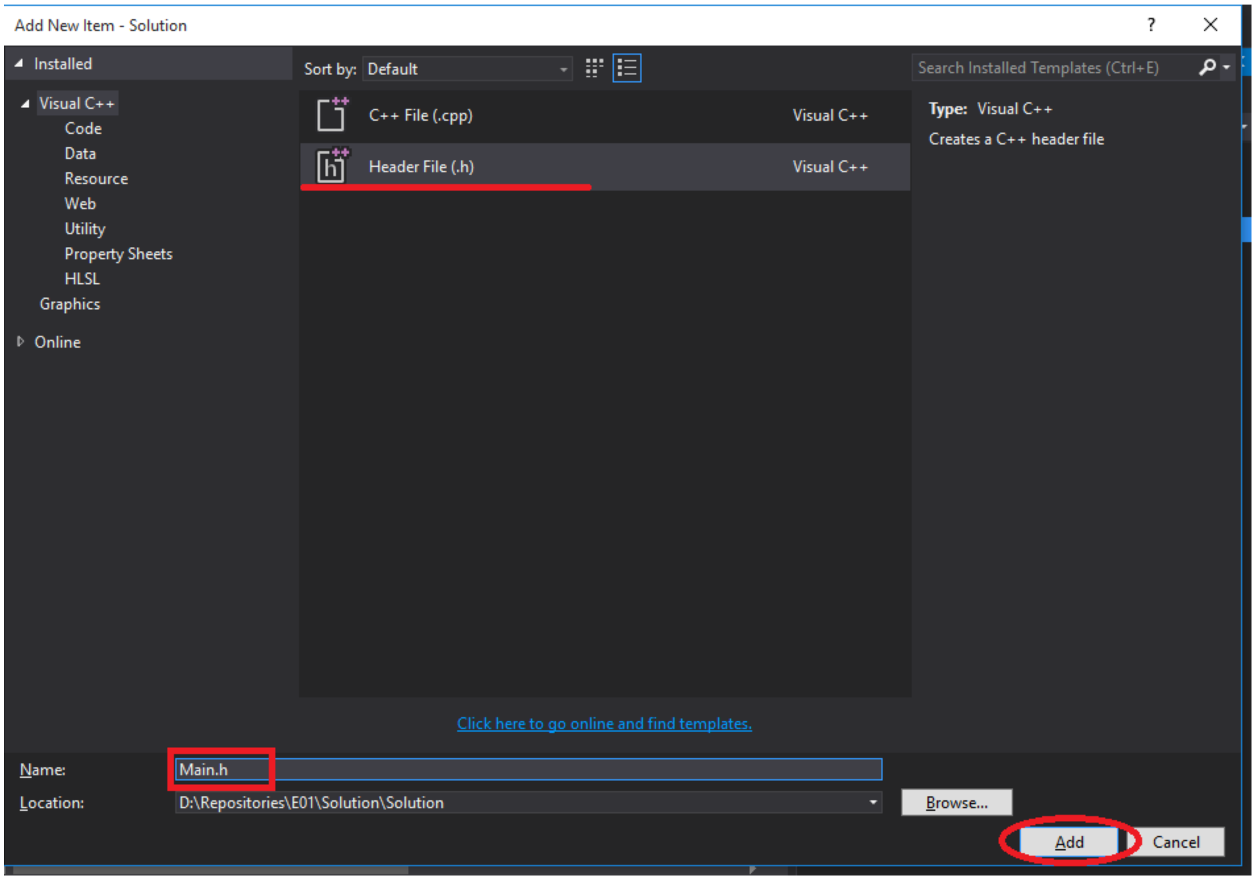
```
        getchar();
        return 0;
}
```

We were unable to automatically populate your Visual Studio Online accounts.

The following error was encountered: TF400813: Resource not available for anonymou

Solution - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug x86 Local Windows Debugger
Luis Alberto Bobadilla Sotelo
Quick Launch (Ctrl+Q)
Main.cpp*
Solution
(Global Scope)
New Item... Ctrl+Shift+A
Existing Item... Shift+Alt+A
New Filter
Class...
Resource...
Spell Checker Configuration File
Add
Class Wizard... Ctrl+Shift+X
Scope to This
New Solution Explorer View
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete Del
Rename
Properties Alt+Enter
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Solution' (1 project)
Solution
External Dependencies
ader Files
ferences
source Files
urce Files
Main.cpp
128 %
Output
Show output from: General
Error List Output
Solution Explorer Team Explorer Class View
This item does not support previewing
Ln 15 Col 1 Ch 1 INS
Server Explorer Toolbox Properties

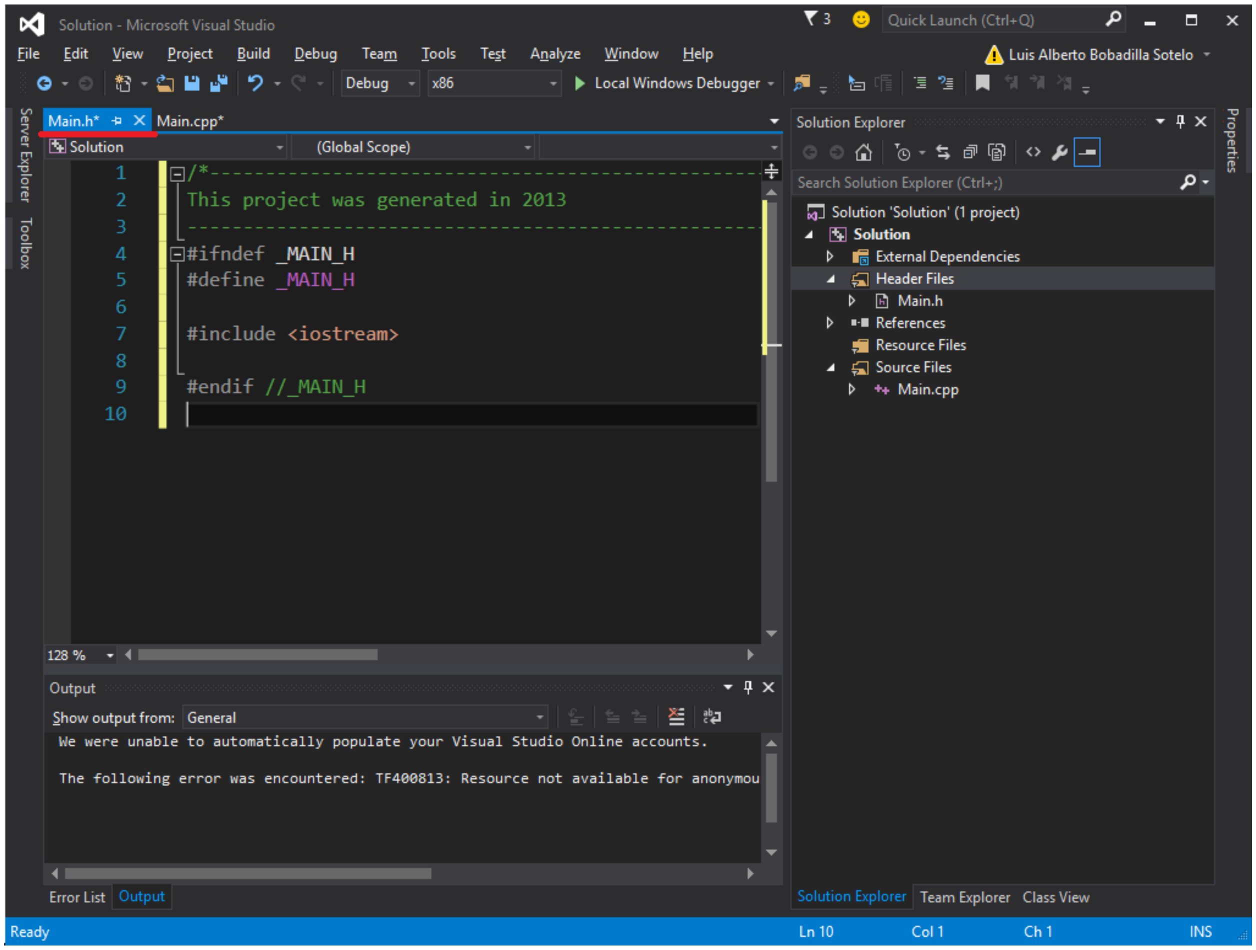
7) Name the file however you want but remember to include the header from the cpp file (Name it "Main" for the sake of this example) and hit OK


Add New Item - Solution
Installed
Visual C++
Code
Data
Resource
Web
Utility
Property Sheets
HLSL
Graphics
Online
Sort by: Default
Search Installed Templates (Ctrl+E)
C++ File (.cpp) Visual C++
Header File (.h) Visual C++
Type: Visual C++
Creates a C++ header file
Click here to go online and find templates.
Browse...
Add
Cancel

Name: Main.h
Location: D:\Repositories\E01\Solution\Solution

8) Create the protectors for your file, so the compiler don't try to compile the file more than once. And include the headers that you will use in your project. (for this example just copy paste these lines,)

```
/*--------------------------------------------------------------------------------
-----------
This project was generated in 2018
--------------------------------------------------------------------------------
---------*/
#ifndef _MAIN_H
#define _MAIN_H

#include <iostream>

#endif //_MAIN_H
```



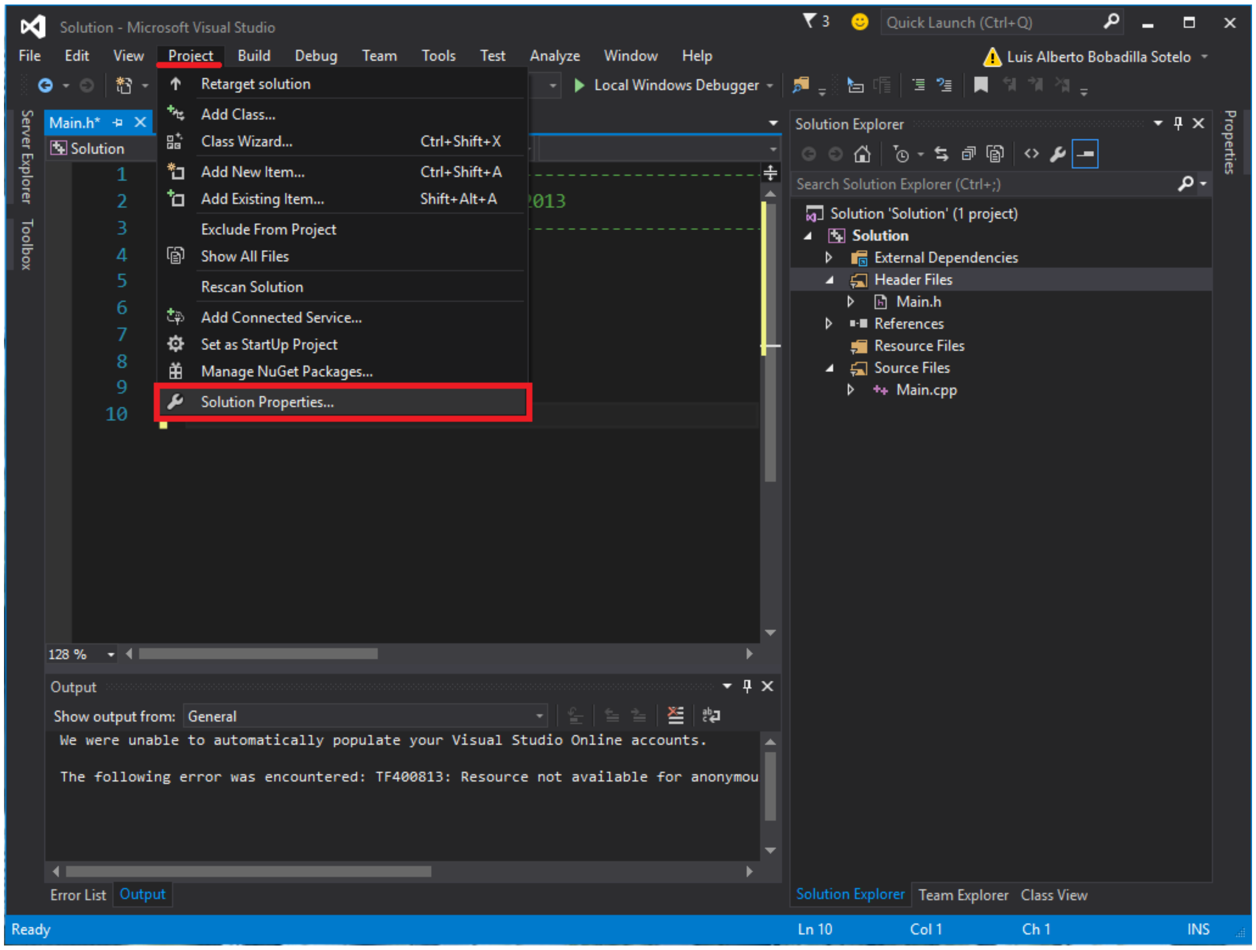
```
/*-----------------------------------------------------------
This project was generated in 2013
-----------------------------------------------------------
#ifndef _MAIN_H
#define _MAIN_H

#include <iostream>

#endif //_MAIN_H
```

```
We were unable to automatically populate your Visual Studio Online accounts.

The following error was encountered: TF400813: Resource not available for anonymou
```

Solution - Microsoft Visual Studio
Quick Launch (Ctrl+Q)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Luis Alberto Bobadilla Sotelo
Debug x86 Local Windows Debugger
Main.h* Main.cpp*
Solution (Global Scope)
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Solution' (1 project)
Solution
External Dependencies
Header Files
Main.h
References
Resource Files
Source Files
Main.cpp
128 %
Output
Show output from: General
Error List Output
Solution Explorer Team Explorer Class View
Ready Ln 10 Col 1 Ch 1 INS

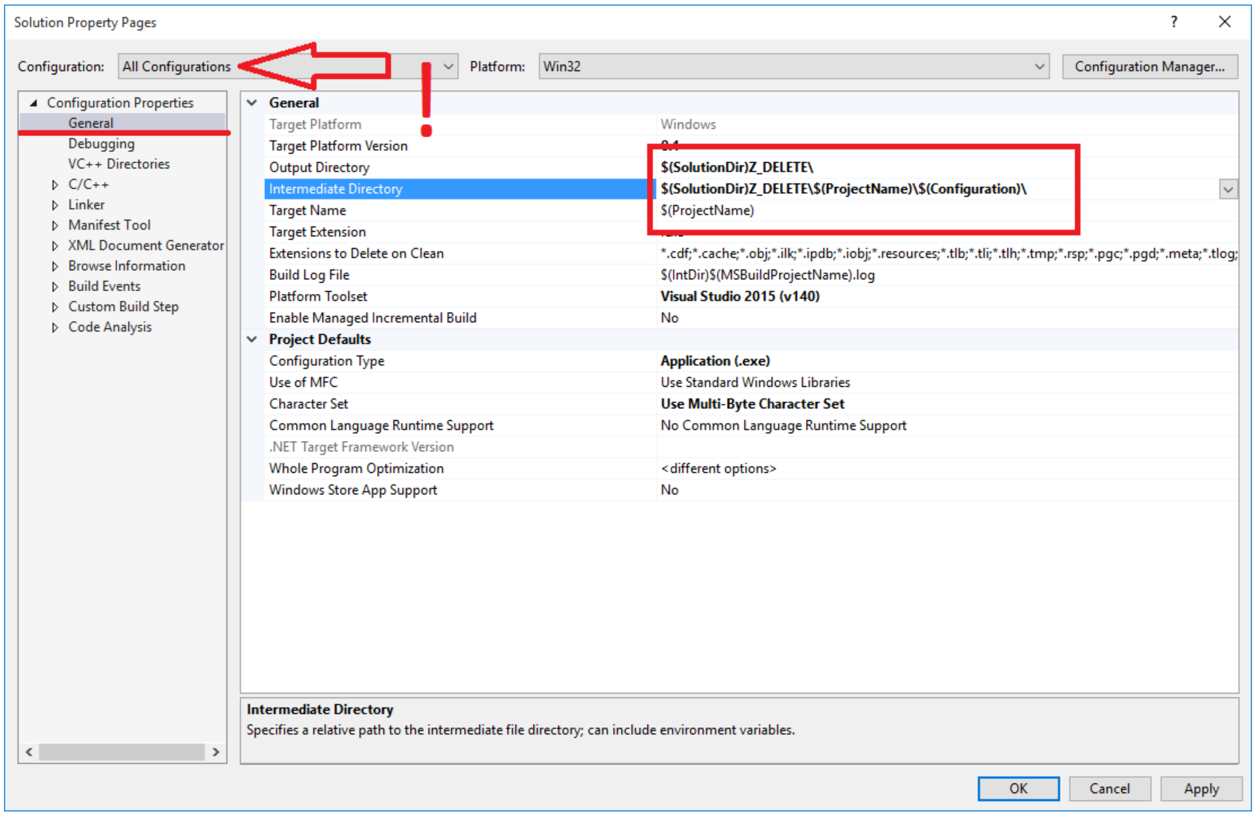
The difference with the #ifndef/#define/#endif and #pragma once is that #pragma is non-standard and it would just work right in most of the compilers but not in all, causing differences in the semantic, while #ifndef/#define/#endif is standard working in all compilers and its usually faster. If you don't get what I tried to say just here… only stick with #ifndef/#define/#endif it's faster in compilation anyway.

9) Go to your project properties. (PROJECT/[Solution] Properties…)



```
We were unable to automatically populate your Visual Studio Online accounts.

The following error was encountered: TF400813: Resource not available for anonymou
```

10) Click on Configuration Properties and make sure you are changing the settings for **all the configurations** go to General and change the Output Directory, the Intermediate Directory and the Target Name fields like follows and apply your changes:
Output Directory: $(SolutionDir)Z_DELETE\
Intermediate Directory: $(SolutionDir)Z_DELETE\$(ProjectName)\$(Configuration)\
Target Name: $(ProjectName)

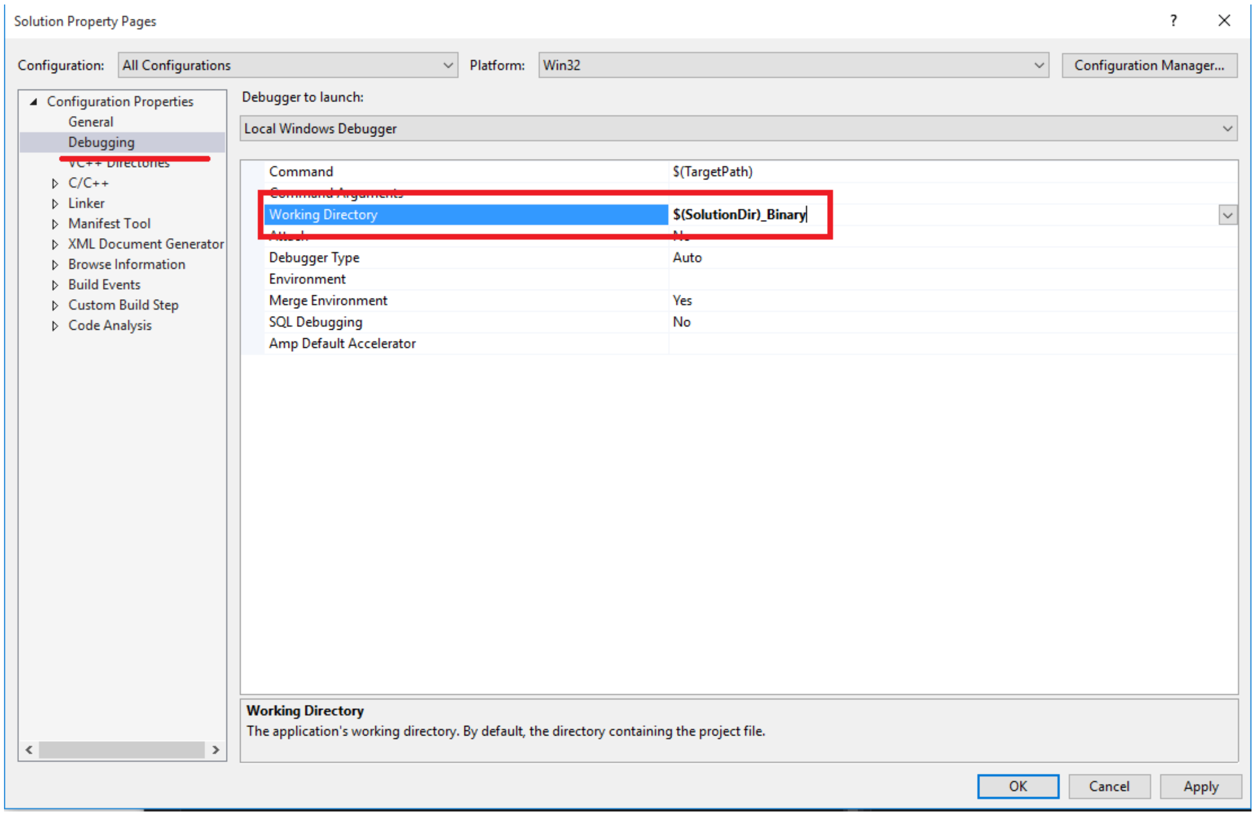


Configuration: All Configurations
Platform: Win32

General	
Target Platform	Windows
Target Platform Version	8.1
Output Directory	**$(SolutionDir)Z_DELETE**
Intermediate Directory	**$(SolutionDir)Z_DELETE\$(ProjectName)\$(Configuration)**
Target Name	$(ProjectName)
Target Extension	.exe
Extensions to Delete on Clean	*.cdf;*.cache;*.obj;*.ilk;*.ipdb;*.iobj;*.resources;*.tlb;*.tli;*.tlh;*.tmp;*.rsp;*.pgc;*.pgd;*.meta;*.tlog;
Build Log File	$(IntDir)$(MSBuildProjectName).log
Platform Toolset	**Visual Studio 2015 (v140)**
Enable Managed Incremental Build	No
Project Defaults	
Configuration Type	**Application (.exe)**
Use of MFC	Use Standard Windows Libraries
Character Set	**Use Multi-Byte Character Set**
Common Language Runtime Support	No Common Language Runtime Support
.NET Target Framework Version	
Whole Program Optimization	<different options>
Windows Store App Support	No

Intermediate Directory
Specifies a relative path to the intermediate file directory; can include environment variables.

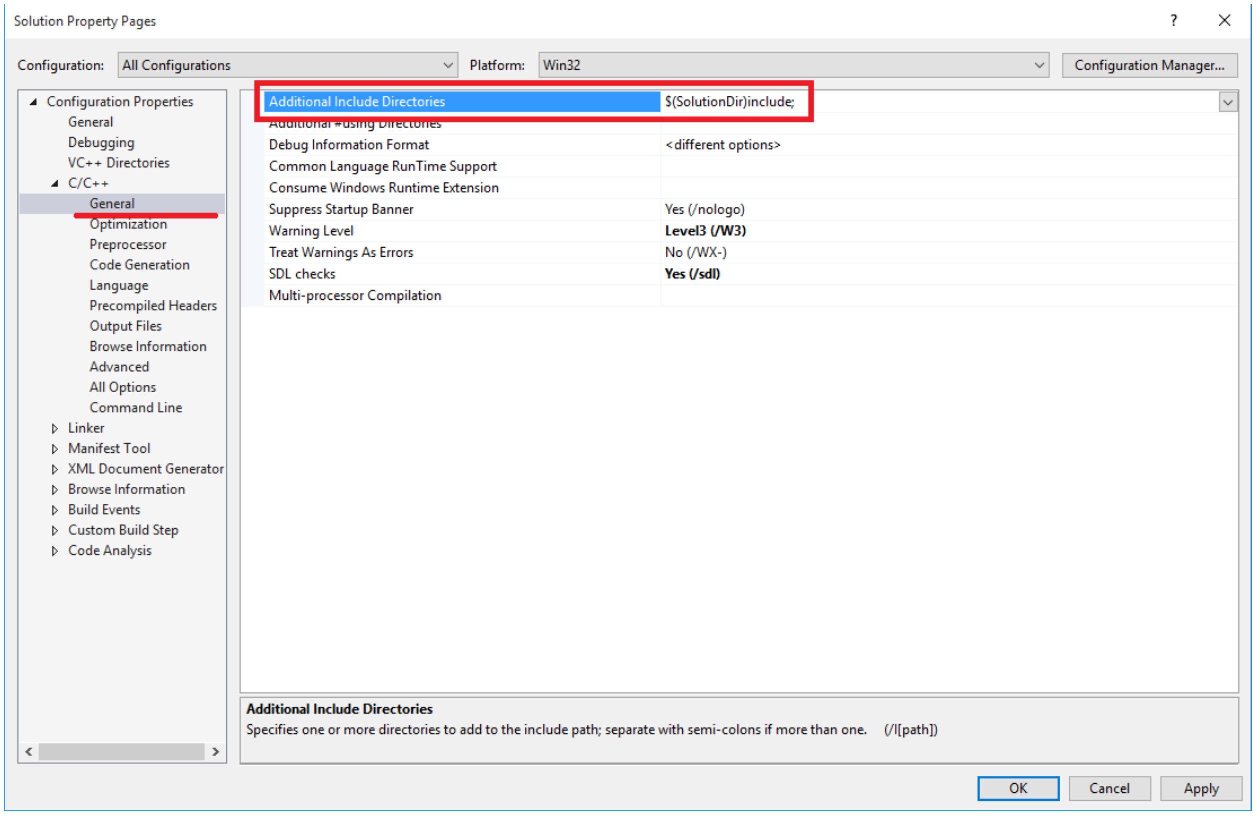
11) In Configuration Properties / Debugging / Working Directory change the field to: $(SolutionDir)_Binary



Command	$(TargetPath)
Command Arguments	
Working Directory	$(SolutionDir)_Binary
Attach	No
Debugger Type	Auto
Environment	
Merge Environment	Yes
SQL Debugging	No
Amp Default Accelerator	

Working Directory
The application's working directory. By default, the directory containing the project file.

12) In C/C++ / General include the folder:
$(SolutionDir)include;

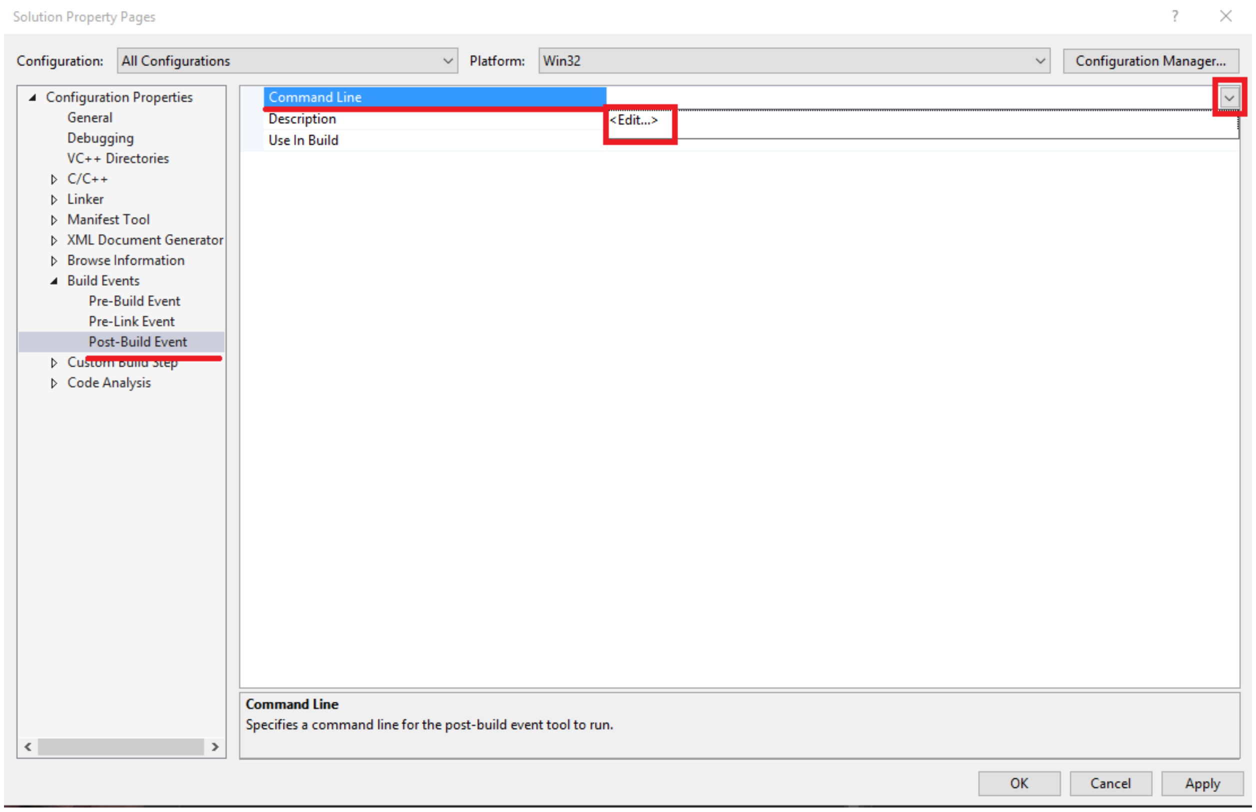


Additional Include Directories	$(SolutionDir)include;
Additional #using Directories	
Debug Information Format	<different options>
Common Language RunTime Support	
Consume Windows Runtime Extension	
Suppress Startup Banner	Yes (/nologo)
Warning Level	**Level3 (/W3)**
Treat Warnings As Errors	No (/WX-)
SDL checks	**Yes (/sdl)**
Multi-processor Compilation	

Additional Include Directories
Specifies one or more directories to add to the include path; separate with semi-colons if more than one. (/I[path])

This way if you have more than one header file that is going to be used among various projects in your solution you don’t need to have extra copies of your headers.

13) In Build Events/ Post-Build Events/ Command Line click on the arrow and <edit…>


Solution Property Pages

Configuration: All Configurations   Platform: Win32   Configuration Manager...

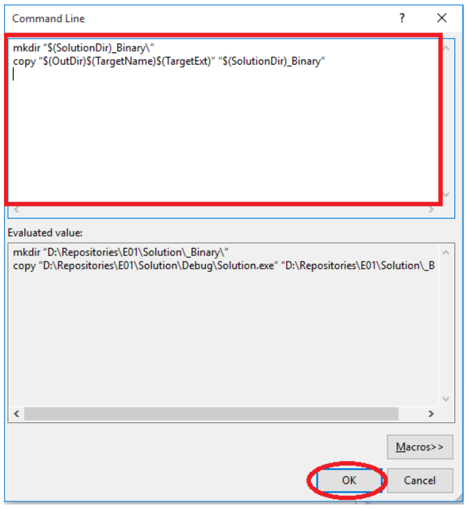
- Configuration Properties
  - General
  - Debugging
  - VC++ Directories
  - C/C++
  - Linker
  - Manifest Tool
  - XML Document Generator
  - Browse Information
  - Build Events
    - Pre-Build Event
    - Pre-Link Event
    - Post-Build Event
  - Custom Build Step
  - Code Analysis

Command Line	<Edit...>
Description	
Use In Build	

Command Line
Specifies a command line for the post-build event tool to run.

OK Cancel Apply

Then add:
mkdir “$(SolutionDir)_Binary\”
copy “$(OutDir)$(TargetName)$(TargetExt)” “$(SolutionDir)_Binary”


Command Line

```
mkdir "$(SolutionDir)_Binary\"
copy "$(OutDir)$(TargetName)$(TargetExt)" "$(SolutionDir)_Binary"
```

Evaluated value:

```
mkdir "D:\Repositories\E01\Solution\_Binary\"
copy "D:\Repositories\E01\Solution\Debug\Solution.exe" "D:\Repositories\E01\Solution\_B
```

Macros>>
OK Cancel

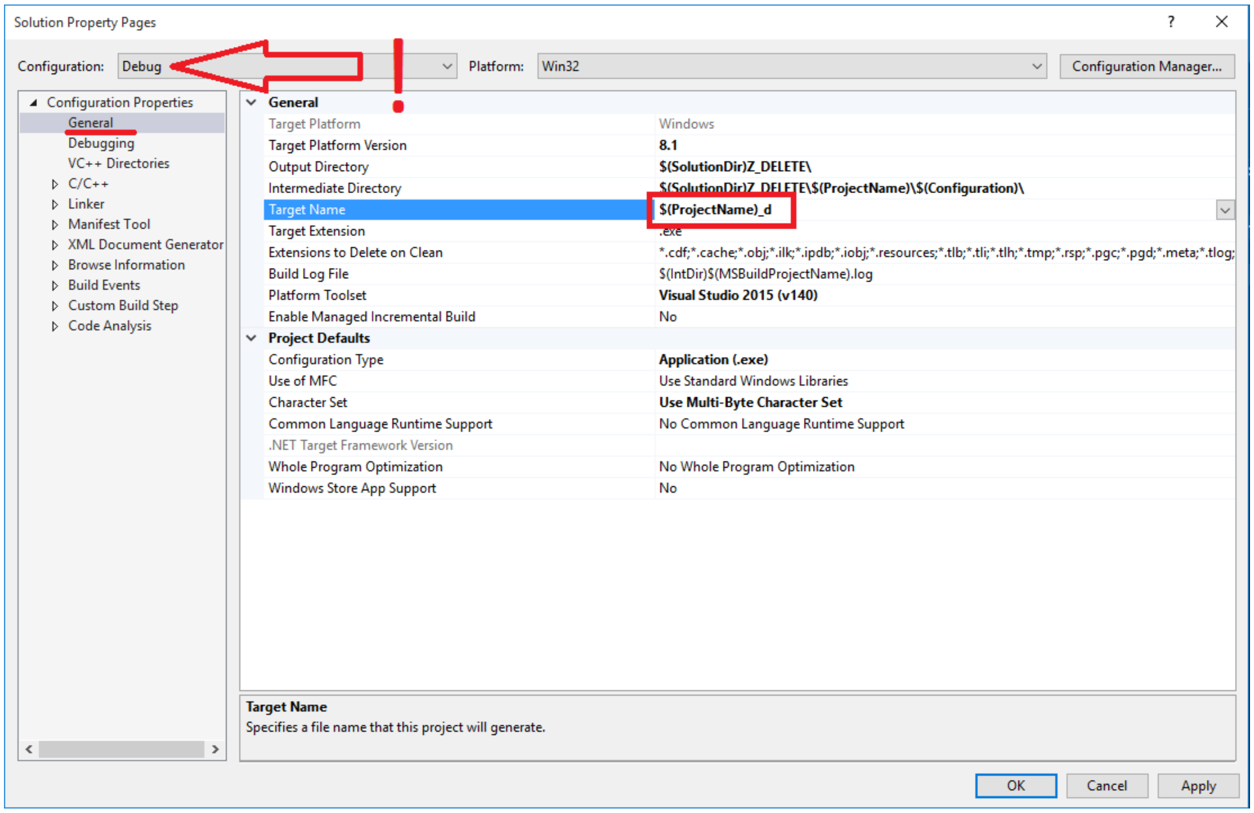
and in Buils Events/ Post-Build Events/ Description add:

Copying binary…

Remember that usually text editors change the “quotes” for text quotes (opening and closing) and they are different symbols than the ones using in most other software be sure that if you are copy-pasting code you replace the quotes by hand or you will have an error.

Apply your changes

14) Change your configuration to **DEBUG** and in Configuration Properties / General add _d at the end of your Target Name and apply the changes.



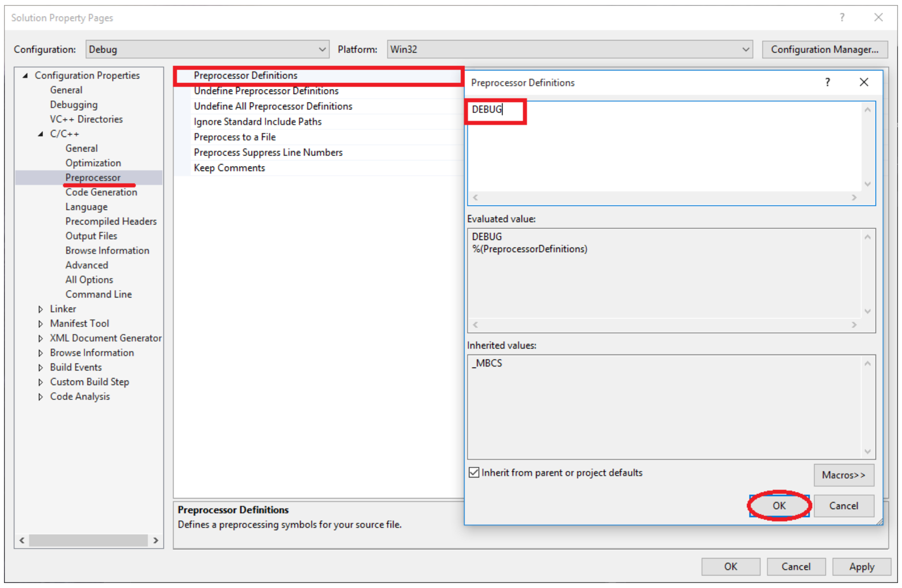
Configuration: Debug   Platform: Win32

- Configuration Properties
  - General
  - Debugging
  - VC++ Directories
  - C/C++
  - Linker
  - Manifest Tool
  - XML Document Generator
  - Browse Information
  - Build Events
  - Custom Build Step
  - Code Analysis

General	
Target Platform	Windows
Target Platform Version	8.1
Output Directory	$(SolutionDir)Z_DELETE\
Intermediate Directory	$(SolutionDir)Z_DELETE\$(ProjectName)\$(Configuration)\
Target Name	$(ProjectName)_d
Target Extension	.exe
Extensions to Delete on Clean	*.cdf;*.cache;*.obj;*.ilk;*.ipdb;*.iobj;*.resources;*.tlb;*.tli;*.tlh;*.tmp;*.rsp;*.pgc;*.pgd;*.meta;*.tlog;
Build Log File	$(IntDir)$(MSBuildProjectName).log
Platform Toolset	Visual Studio 2015 (v140)
Enable Managed Incremental Build	No
Project Defaults	
Configuration Type	Application (.exe)
Use of MFC	Use Standard Windows Libraries
Character Set	Use Multi-Byte Character Set
Common Language Runtime Support	No Common Language Runtime Support
.NET Target Framework Version	
Whole Program Optimization	No Whole Program Optimization
Windows Store App Support	No

Target Name
Specifies a file name that this project will generate.

15) Still in Debug configuration in C/C++ / Preprocessor/ Preprocessor Definitions click on the arrow and add DEBUG


Solution Property Pages

Configuration: Debug   Platform: Win32   Configuration Manager...

Configuration Properties
- General
- Debugging
- VC++ Directories
- C/C++
  - General
  - Optimization
  - Preprocessor
  - Code Generation
  - Language
  - Precompiled Headers
  - Output Files
  - Browse Information
  - Advanced
  - All Options
  - Command Line
- Linker
- Manifest Tool
- XML Document Generator
- Browse Information
- Build Events
- Custom Build Step
- Code Analysis

Preprocessor Definitions
Undefine Preprocessor Definitions
Undefine All Preprocessor Definitions
Ignore Standard Include Paths
Preprocess to a File
Preprocess Suppress Line Numbers
Keep Comments

Preprocessor Definitions
Defines a preprocessing symbols for your source file.

Preprocessor Definitions

```
DEBUG
```

Evaluated value:

```
DEBUG
%(PreprocessorDefinitions)
```

Inherited values:

```
_MBCS
```

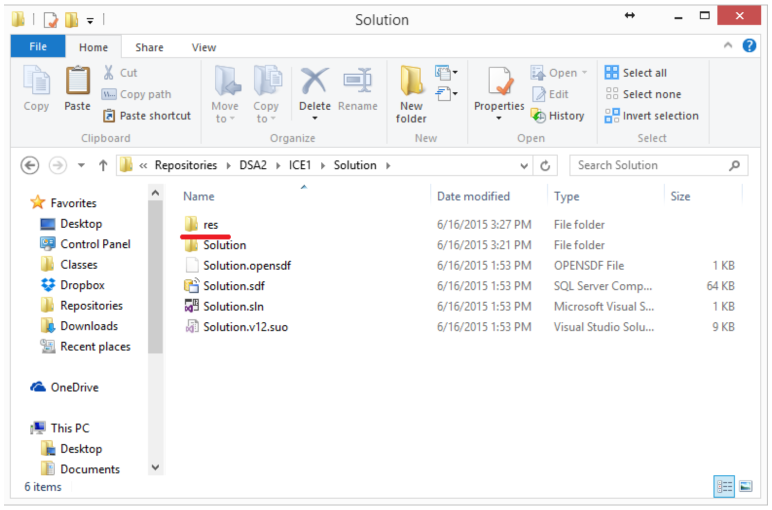
☑ Inherit from parent or project defaults   Macros>>

OK   Cancel

OK   Cancel   Apply

With this you are defining DEBUG as a keyword for your entire project.
With this configuration you will add all the intermediary and binary files to a folder called Z_DELETE at the same level as your solution and will copy the binary to a new folder called _Binary.
Hit OK to close the configuration now

16) Copy the provided res folder at the same level of your sln file:

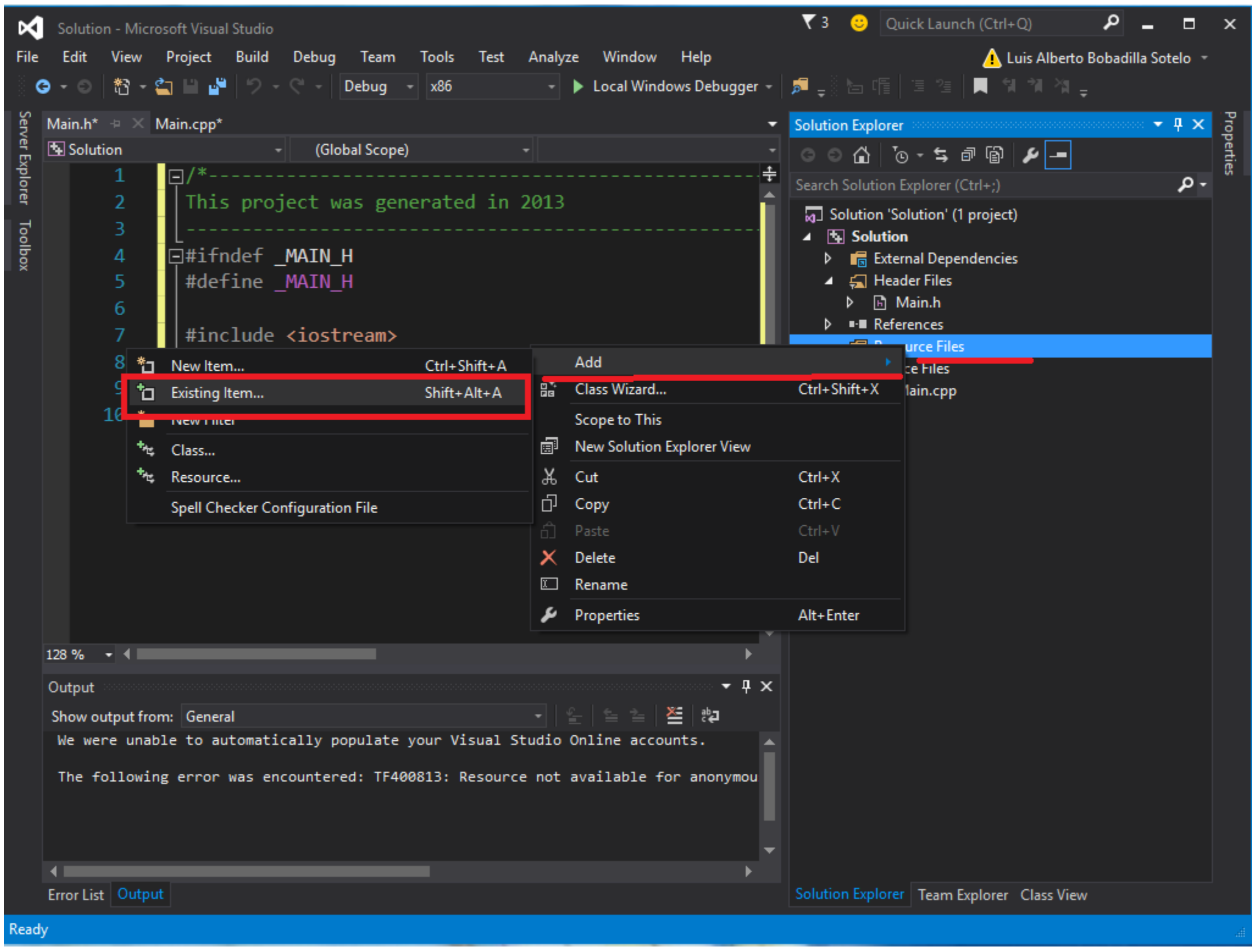

Solution

« Repositories › DSA2 › ICE1 › Solution ›   Search Solution

Name	Date modified	Type	Size
res	6/16/2015 3:27 PM	File folder	
Solution	6/16/2015 3:21 PM	File folder	
Solution.opensdf	6/16/2015 1:53 PM	OPENSDF File	1 KB
Solution.sdf	6/16/2015 1:53 PM	SQL Server Comp...	64 KB
Solution.sln	6/16/2015 1:53 PM	Microsoft Visual S...	1 KB
Solution.v12.suo	6/16/2015 1:53 PM	Visual Studio Solu...	9 KB

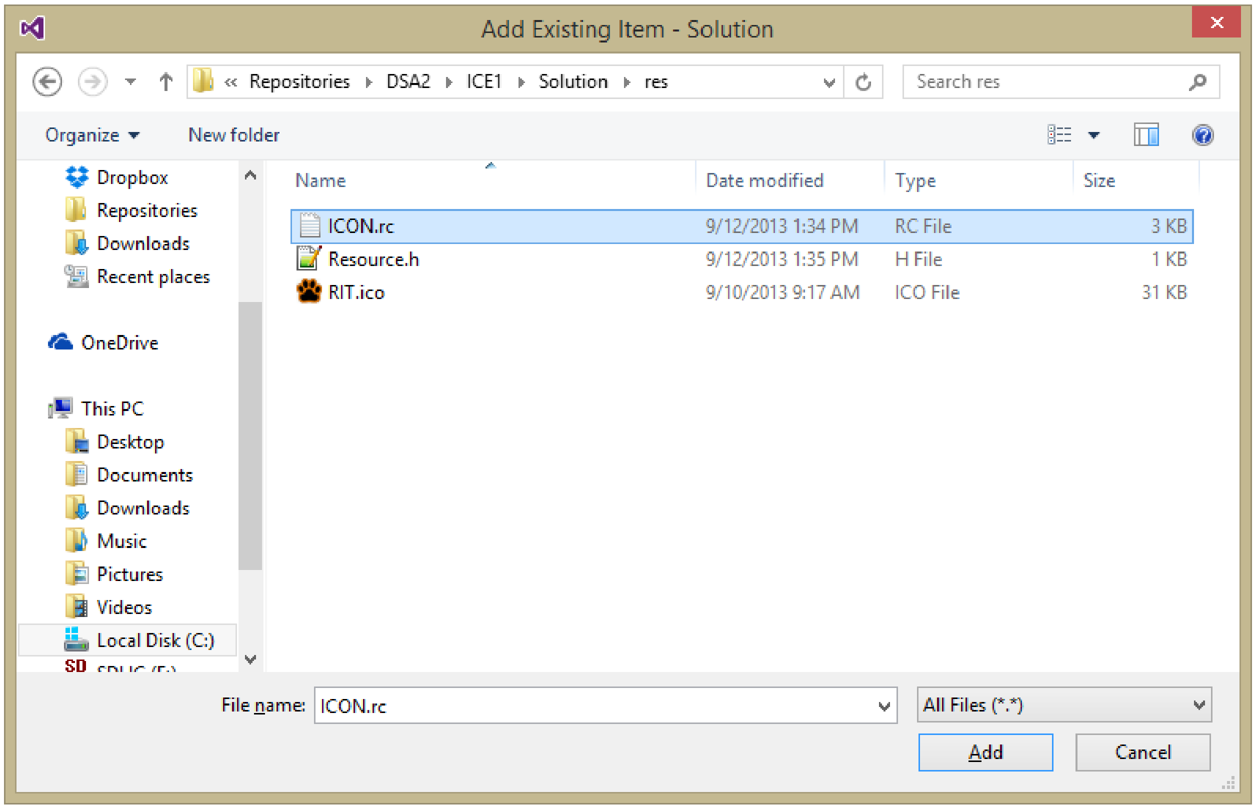
6 items

17) Add the ICON.rc into the RESOURCES FOLDER (right-click Resources Files/Add/Existing Item)



```
/*-------------------------------------------------------------
This project was generated in 2013
-------------------------------------------------------------
#ifndef _MAIN_H
#define _MAIN_H

#include <iostream>
```

```
We were unable to automatically populate your Visual Studio Online accounts.

The following error was encountered: TF400813: Resource not available for anonymou
```



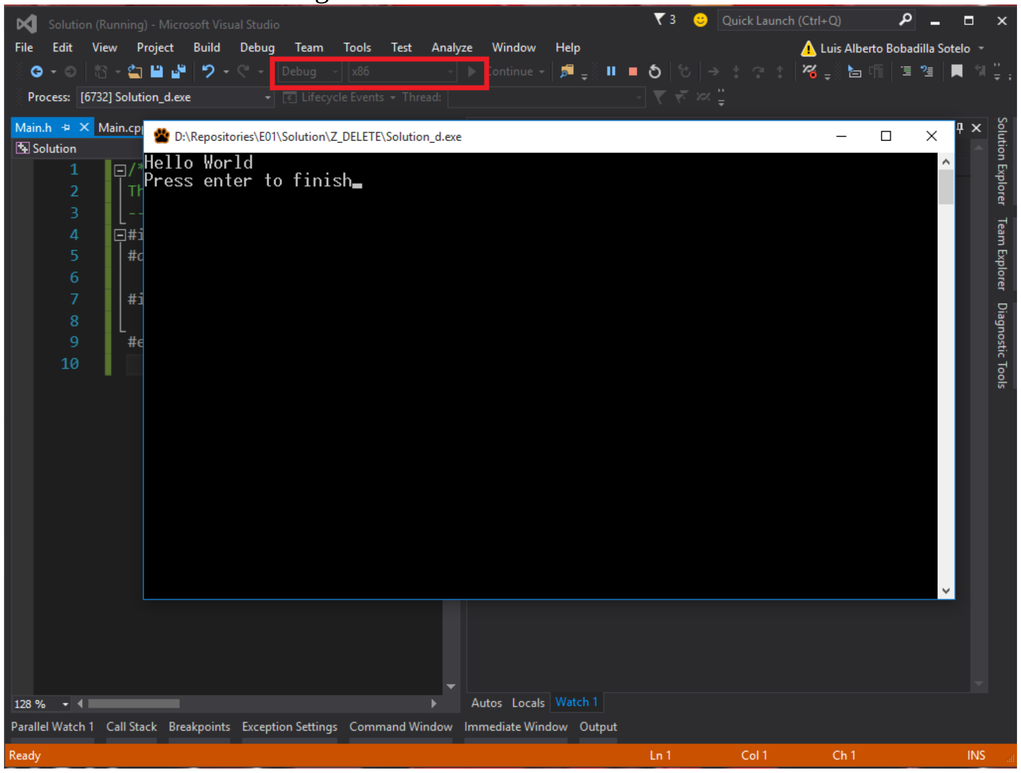
Name	Date modified	Type	Size
ICON.rc	9/12/2013 1:34 PM	RC File	3 KB
Resource.h	9/12/2013 1:35 PM	H File	1 KB
RIT.ico	9/10/2013 9:17 AM	ICO File	31 KB

File name: ICON.rc

All Files (*.*)

This will add an Icon to your application, if you ever want to add a different icon to your project you just need to specify said icon in the ICON.rc file accordingly (You can edit the file in Note Pad, or the icon directly in VS).

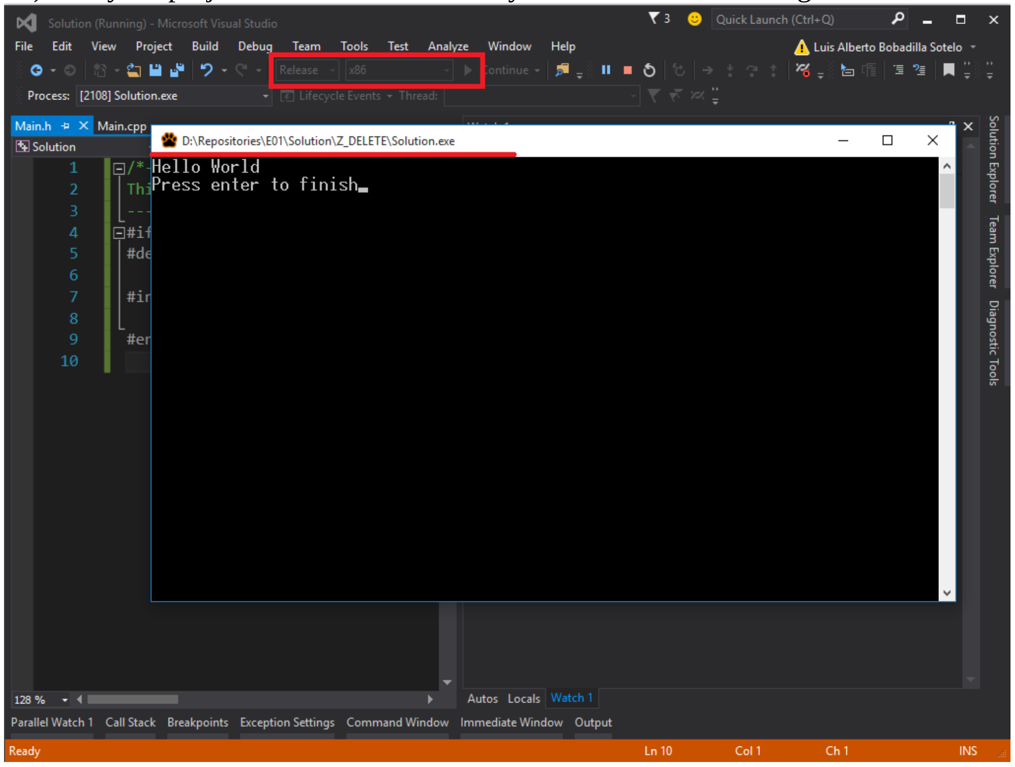
18) Run your project in Debug mode (with F5 only, not break-debugging it with Control+F5) and you should see something like this:


Solution (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug
x86
Process: [6732] Solution_d.exe
D:\Repositories\E01\Solution\Z_DELETE\Solution_d.exe

```
Hello World
Press enter to finish_
```

As you can see the name of the executable ends in _d

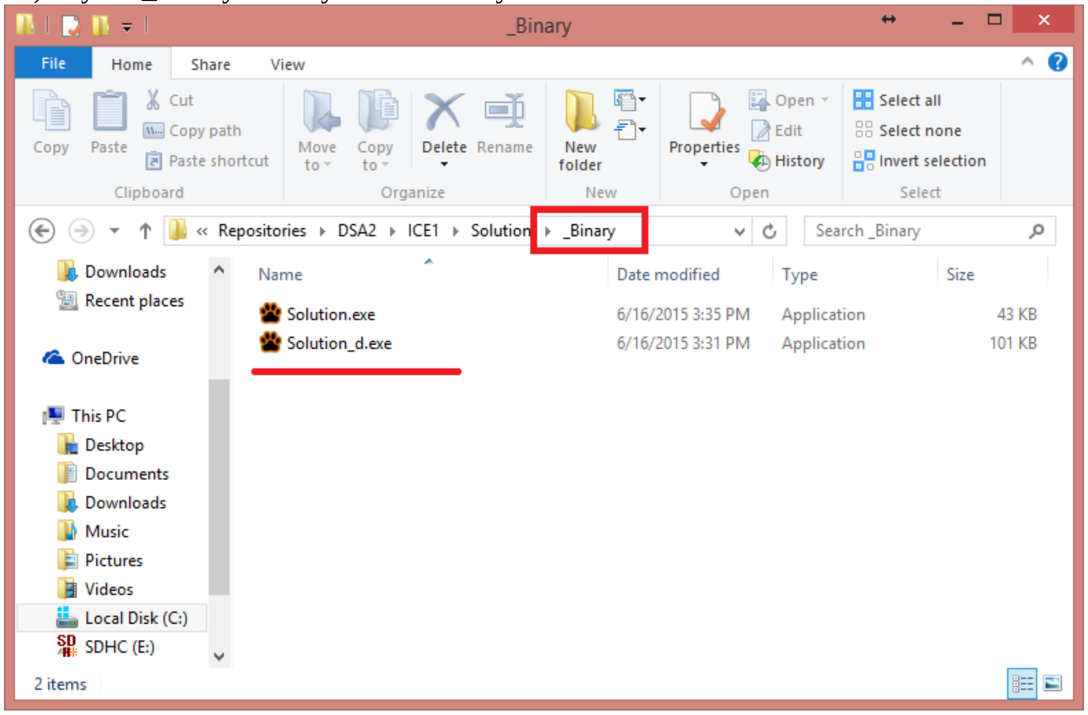
19) Run your project in Release mode and you should see something like this:


Solution (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Release
x86
Process: [2108] Solution.exe
D:\Repositories\E01\Solution\Z_DELETE\Solution.exe

```
Hello World
Press enter to finish_
```

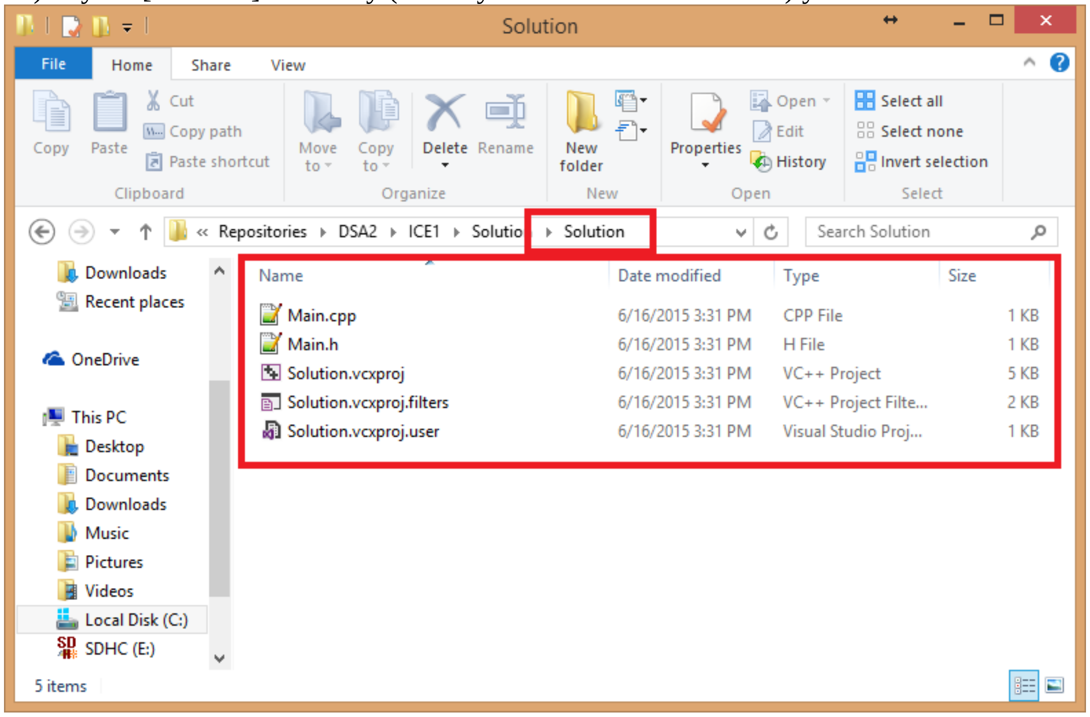
The executable should be missing the _d now

20) in your _Binary folder you should only have two files:



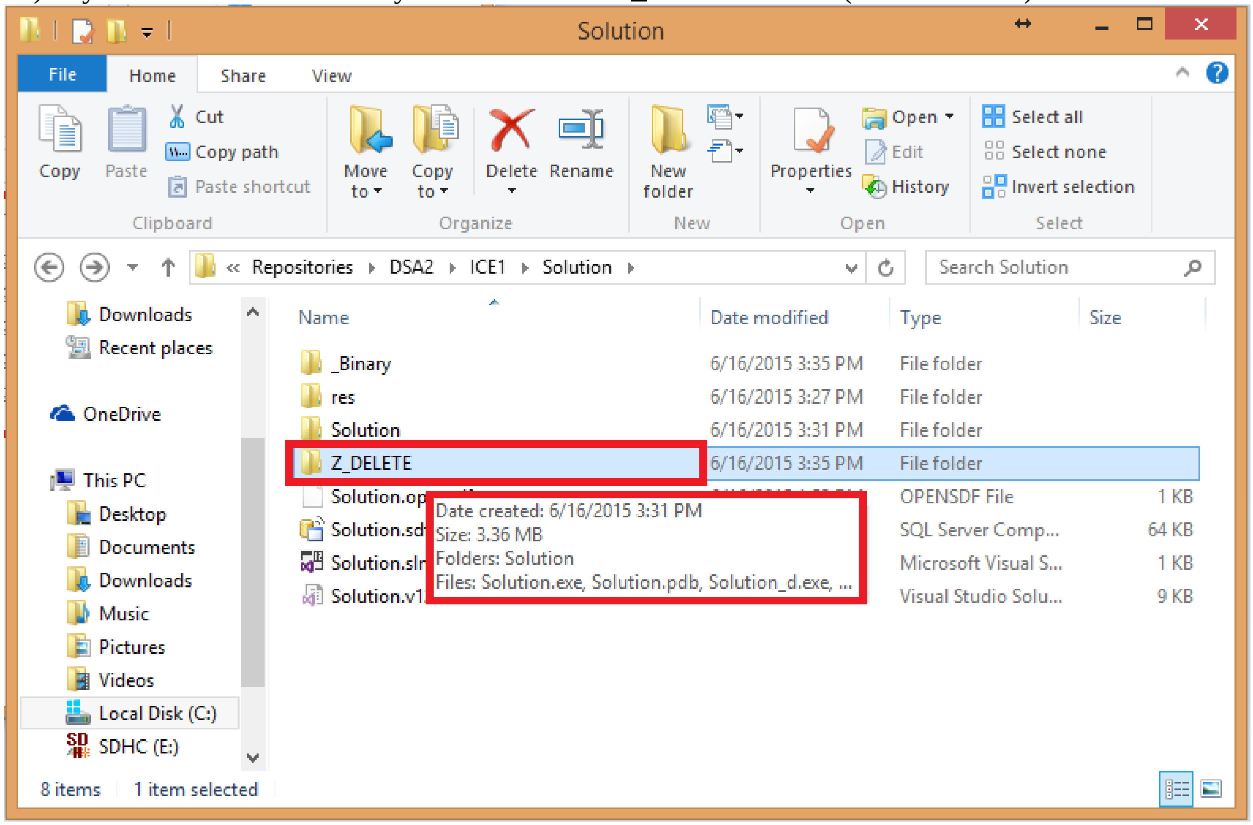
Name	Date modified	Type	Size
Solution.exe	6/16/2015 3:35 PM	Application	43 KB
Solution_d.exe	6/16/2015 3:31 PM	Application	101 KB

21) in your [Solution] Directory (where your c++s and headers are) you should have no subfolders:



Name	Date modified	Type	Size
Main.cpp	6/16/2015 3:31 PM	CPP File	1 KB
Main.h	6/16/2015 3:31 PM	H File	1 KB
Solution.vcxproj	6/16/2015 3:31 PM	VC++ Project	5 KB
Solution.vcxproj.filters	6/16/2015 3:31 PM	VC++ Project Filte...	2 KB
Solution.vcxproj.user	6/16/2015 3:31 PM	Visual Studio Proj...	1 KB

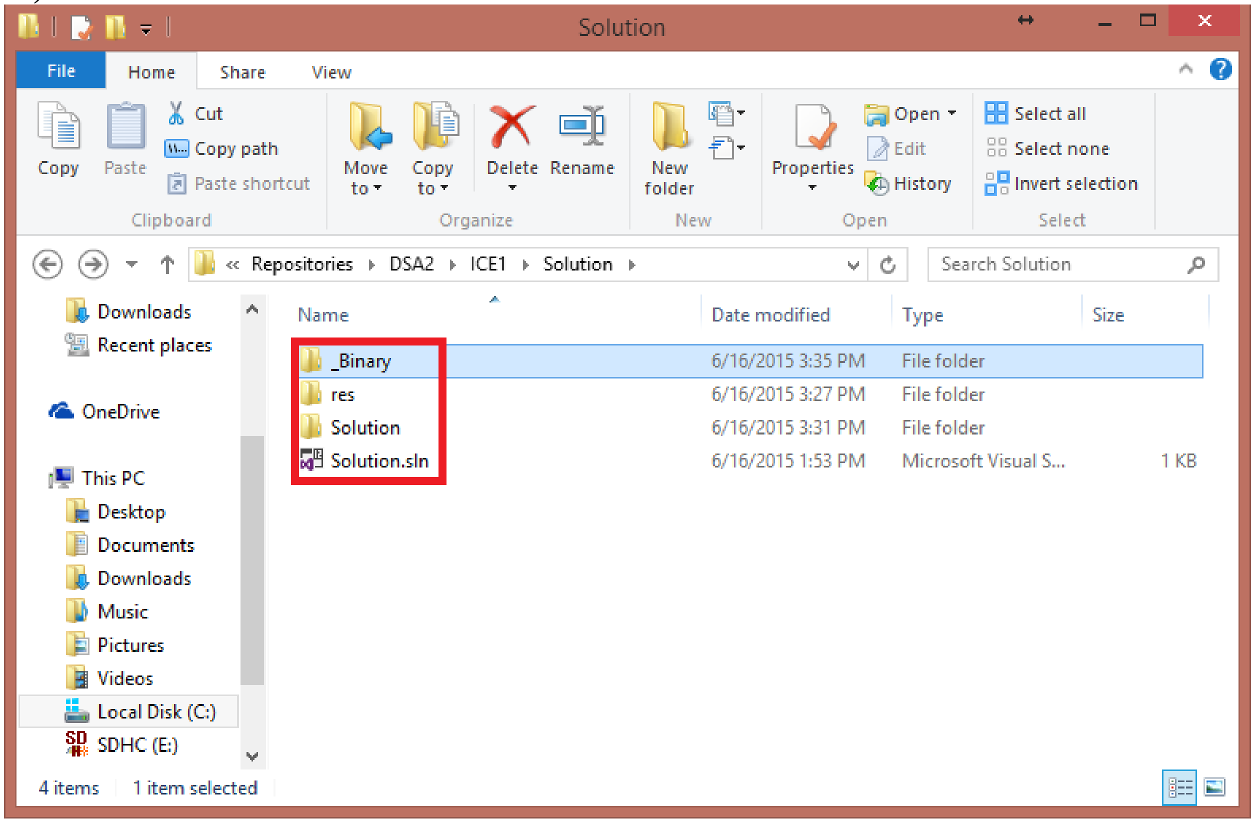
22) In your main Solution folder you should have a Z_DELETE folder (of about 3 MB)



Name	Date modified	Type	Size
_Binary	6/16/2015 3:35 PM	File folder	
res	6/16/2015 3:27 PM	File folder	
Solution	6/16/2015 3:31 PM	File folder	
Z_DELETE	6/16/2015 3:35 PM	File folder	
Solution.op		OPENSDF File	1 KB
Solution.sd		SQL Server Comp...	64 KB
Solution.sln		Microsoft Visual S...	1 KB
Solution.v1		Visual Studio Solu...	9 KB

Date created: 6/16/2015 3:31 PM
Size: 3.36 MB
Folders: Solution
Files: Solution.exe, Solution.pdb, Solution_d.exe, ...

23) Close visual studio and delete all the trash files:



Name	Date modified	Type	Size
_Binary	6/16/2015 3:35 PM	File folder	
res	6/16/2015 3:27 PM	File folder	
Solution	6/16/2015 3:31 PM	File folder	
Solution.sln	6/16/2015 1:53 PM	Microsoft Visual S...	1 KB

24) You should only have a Solution.sln file and those 3 folders (and maybe a hidden .suo file do not worry about that file).

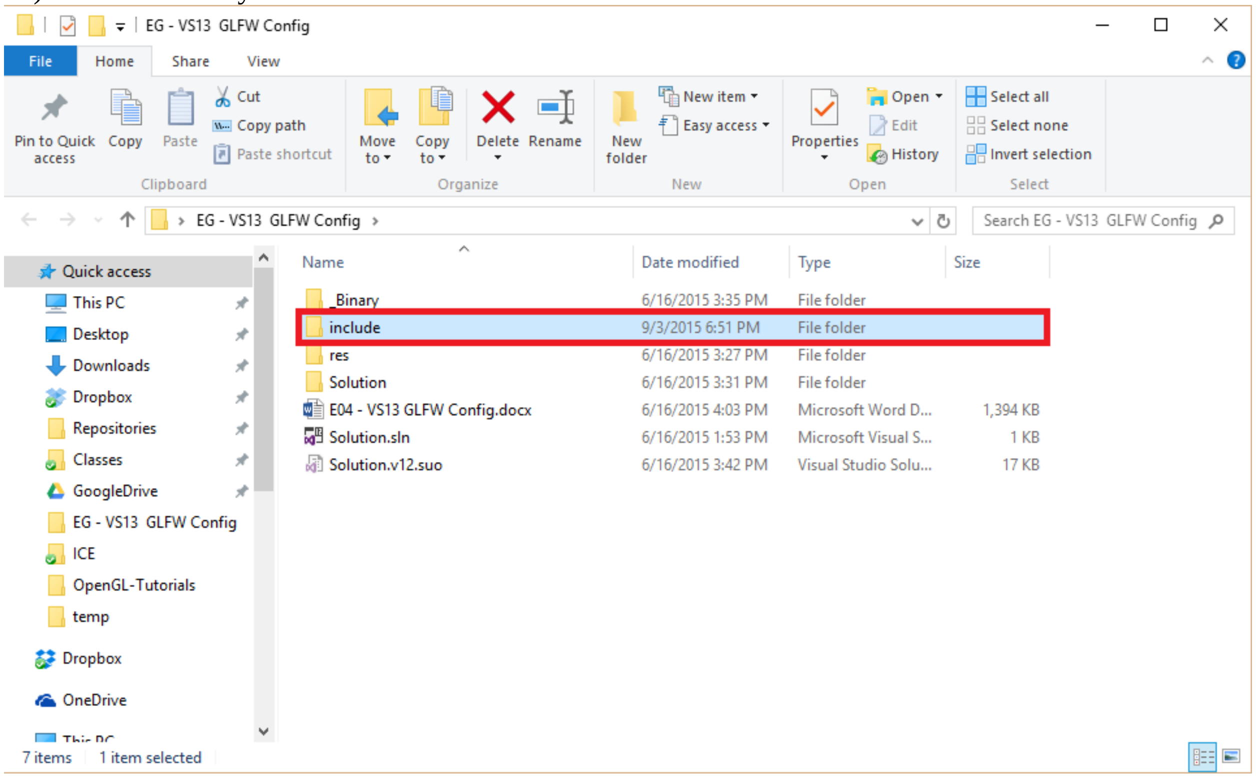
When you open an .sln file Visual studio is going to recreate the data base definition for your variables and functions (the .sdf file) this file is used inside your configuration for your machine only if you share this project the new computer will get rid of this file and create a new tailored one, thus making it completely useless for sharing. It also uses A LOT of space.

Inside of your Z_DELETE folder visual studio will create the final binary and all of the intermediary files that are necessary in order to make those, again all these files are recreated by Visual Studio as needed, thus you are completely safe by just deleting this folder before you zip your project and share it.

A zipped version of this project is going to use as much as 3 MB without all the "trash files" a project that has no "trash files" is roughly around 100KB, this is why GETTING RID OF ALL THE TRASH FILES IS A REQUIREMENT FOR ANY SUBMISSION FOR THIS CLASS.

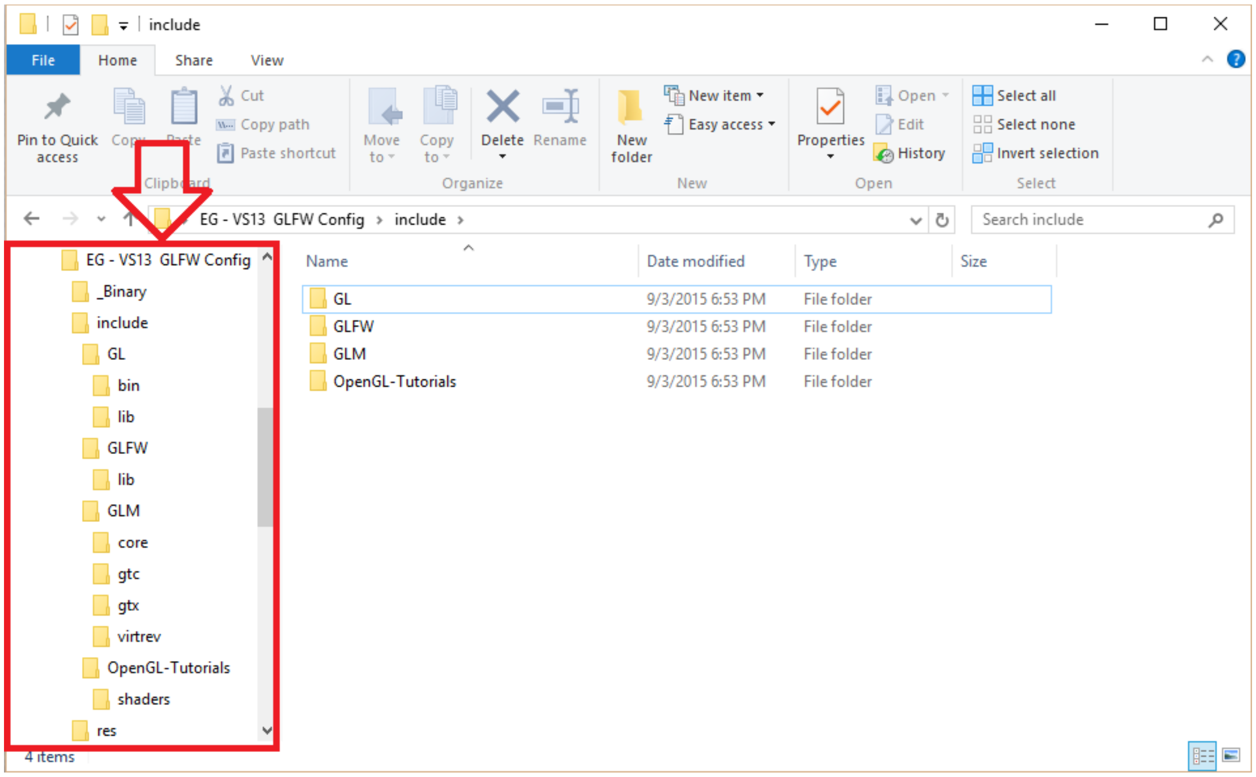
NEW INSTRUCTIONS

25) In the root of your solution create an include folder:



Name	Date modified	Type	Size
_Binary	6/16/2015 3:35 PM	File folder	
include	9/3/2015 6:51 PM	File folder	
res	6/16/2015 3:27 PM	File folder	
Solution	6/16/2015 3:31 PM	File folder	
E04 - VS13 GLFW Config.docx	6/16/2015 4:03 PM	Microsoft Word D...	1,394 KB
Solution.sln	6/16/2015 1:53 PM	Microsoft Visual S...	1 KB
Solution.v12.suo	6/16/2015 3:42 PM	Visual Studio Solu...	17 KB

26) Inside of that folder unzip the provided library files in such a way they follow the folder structure on the left:



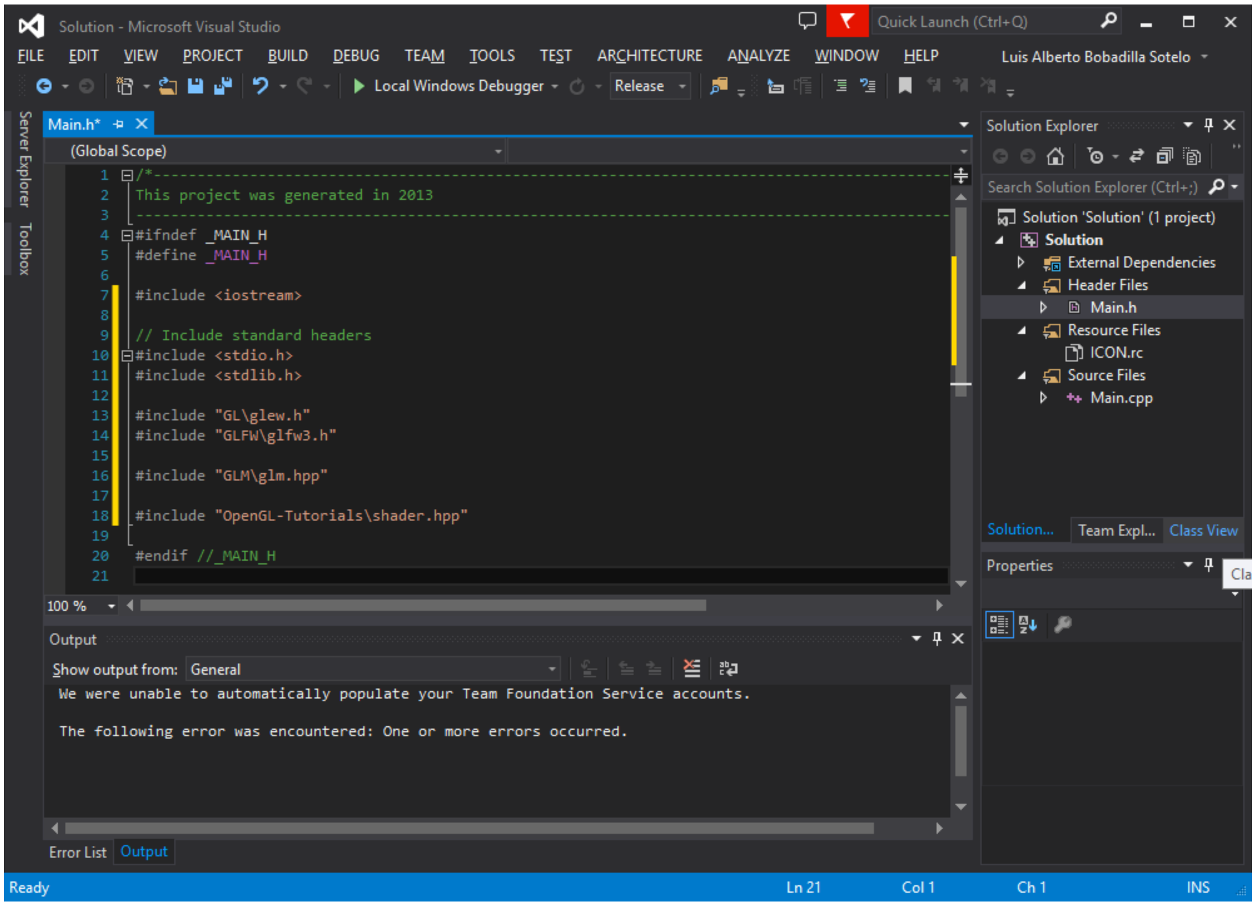
Folder tree:

- EG - VS13 GLFW Config
  - _Binary
  - include
    - GL
      - bin
      - lib
    - GLFW
      - lib
    - GLM
      - core
      - gtc
      - gtx
      - virtrev
    - OpenGL-Tutorials
      - shaders
  - res

Name	Date modified	Type	Size
GL	9/3/2015 6:53 PM	File folder	
GLFW	9/3/2015 6:53 PM	File folder	
GLM	9/3/2015 6:53 PM	File folder	
OpenGL-Tutorials	9/3/2015 6:53 PM	File folder	

27) in your main.h include the following after <iostream>:

```
// Include standard headers
#include <stdio.h>
#include <stdlib.h>

#include "GL\glew.h"
#include "GLFW\glfw3.h"

#include "GLM\glm.hpp"

#include "OpenGL-Tutorials\shader.hpp"
```

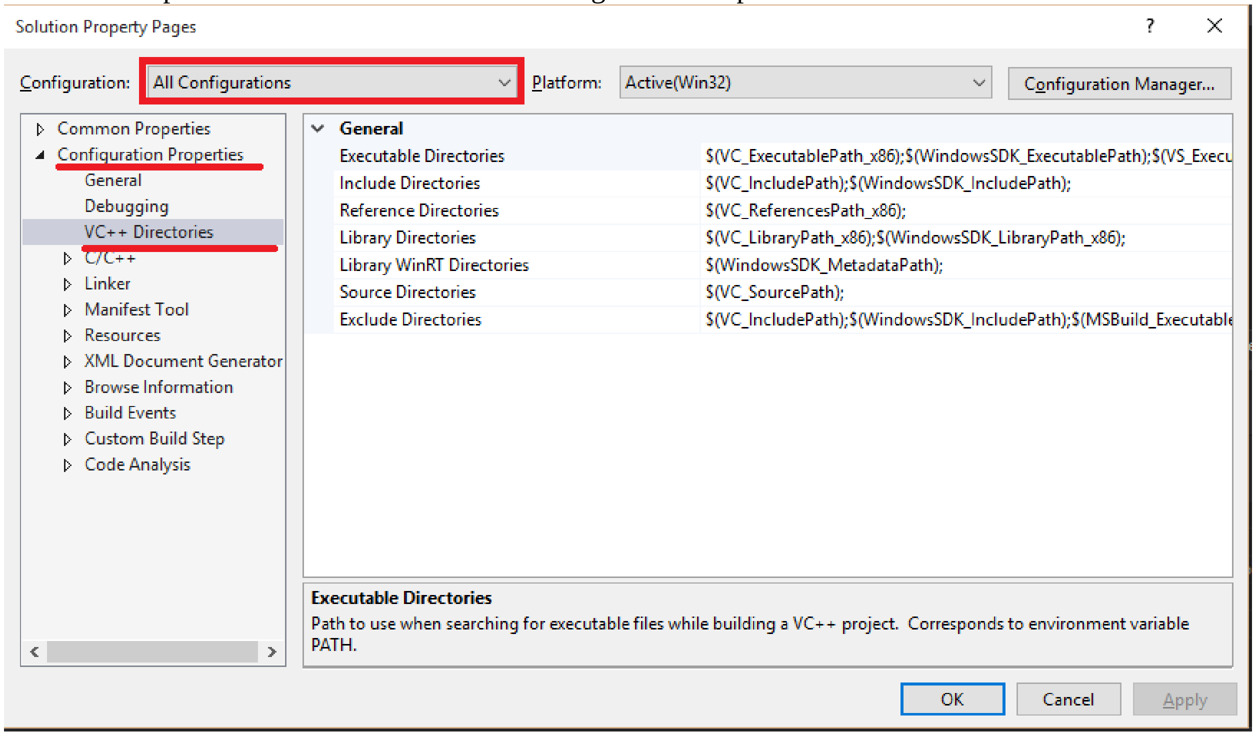


```
/*------------------------------------------------------------------------
This project was generated in 2013
------------------------------------------------------------------------*/
#ifndef _MAIN_H
#define _MAIN_H

#include <iostream>

// Include standard headers
#include <stdio.h>
#include <stdlib.h>

#include "GL\glew.h"
#include "GLFW\glfw3.h"

#include "GLM\glm.hpp"

#include "OpenGL-Tutorials\shader.hpp"

#endif //_MAIN_H
```

```
We were unable to automatically populate your Team Foundation Service accounts.

The following error was encountered: One or more errors occurred.
```

With that you are telling the compiler where the libraries are going to live under your project, the issue is they are relative routes right now, and the compiler ignores the real location of these files, we will fix that next.

28) Go to the properties of your project again (make sure you are under “all configurations” settings) and look for the option VC++ Directories under Configuration Properties:



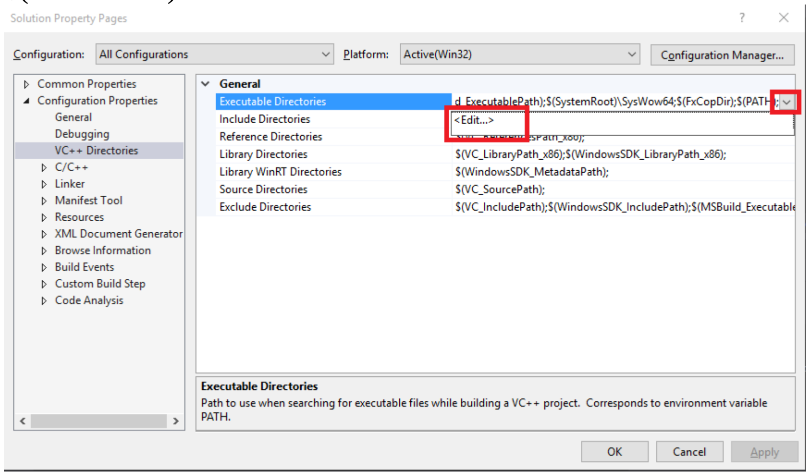
Solution Property Pages

Configuration: All Configurations
Platform: Active(Win32)

General	
Executable Directories	$(VC_ExecutablePath_x86);$(WindowsSDK_ExecutablePath);$(VS_Execu
Include Directories	$(VC_IncludePath);$(WindowsSDK_IncludePath);
Reference Directories	$(VC_ReferencesPath_x86);
Library Directories	$(VC_LibraryPath_x86);$(WindowsSDK_LibraryPath_x86);
Library WinRT Directories	$(WindowsSDK_MetadataPath);
Source Directories	$(VC_SourcePath);
Exclude Directories	$(VC_IncludePath);$(WindowsSDK_IncludePath);$(MSBuild_Executable

Executable Directories
Path to use when searching for executable files while building a VC++ project. Corresponds to environment variable PATH.

29 a) making sure you edit the properties and not just pasting this addresses (using the arrows) add the following folders:
$(SolutionDir)include\GL\bin

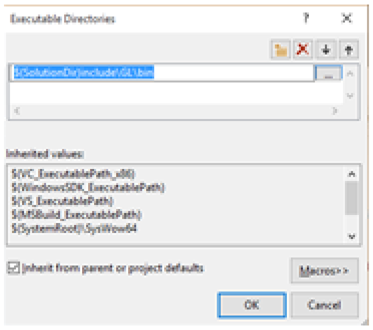


Solution Property Pages

Configuration: All Configurations
Platform: Active(Win32)

General	
Executable Directories	d_ExecutablePath);$(SystemRoot)\SysWow64;$(FxCopDir);$(PATH);
Include Directories	<Edit...>
Reference Directories	
Library Directories	$(VC_LibraryPath_x86);$(WindowsSDK_LibraryPath_x86);
Library WinRT Directories	$(WindowsSDK_MetadataPath);
Source Directories	$(VC_SourcePath);
Exclude Directories	$(VC_IncludePath);$(WindowsSDK_IncludePath);$(MSBuild_Executable

Executable Directories
Path to use when searching for executable files while building a VC++ project. Corresponds to environment variable PATH.



Executable Directories

$(SolutionDir)include\GL\bin

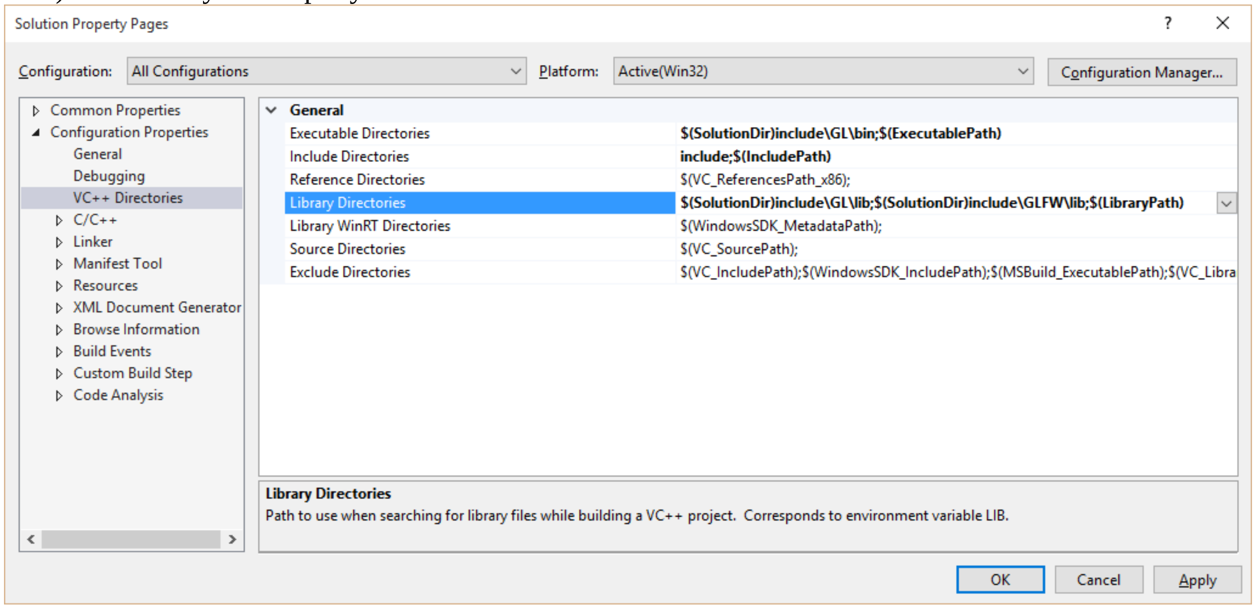
Inherited values:
$(VC_ExecutablePath_x86)
$(WindowsSDK_ExecutablePath)
$(VS_ExecutablePath)
$(MSBuild_ExecutablePath)
$(SystemRoot)\SysWow64

[✓] Inherit from parent or project defaults

29 b) Do the same for Include directories but add the following instead:
include

29 c) For Library directories add:
$(SolutionDir)include\GL\lib
$(SolutionDir)include\GLFW\lib
-make sure it looks like this in the end:
$(SolutionDir)include\GL\lib;$(SolutionDir)include\GLFW\lib;$(LibraryPath)

29 d) At the end your Property window should look like this:

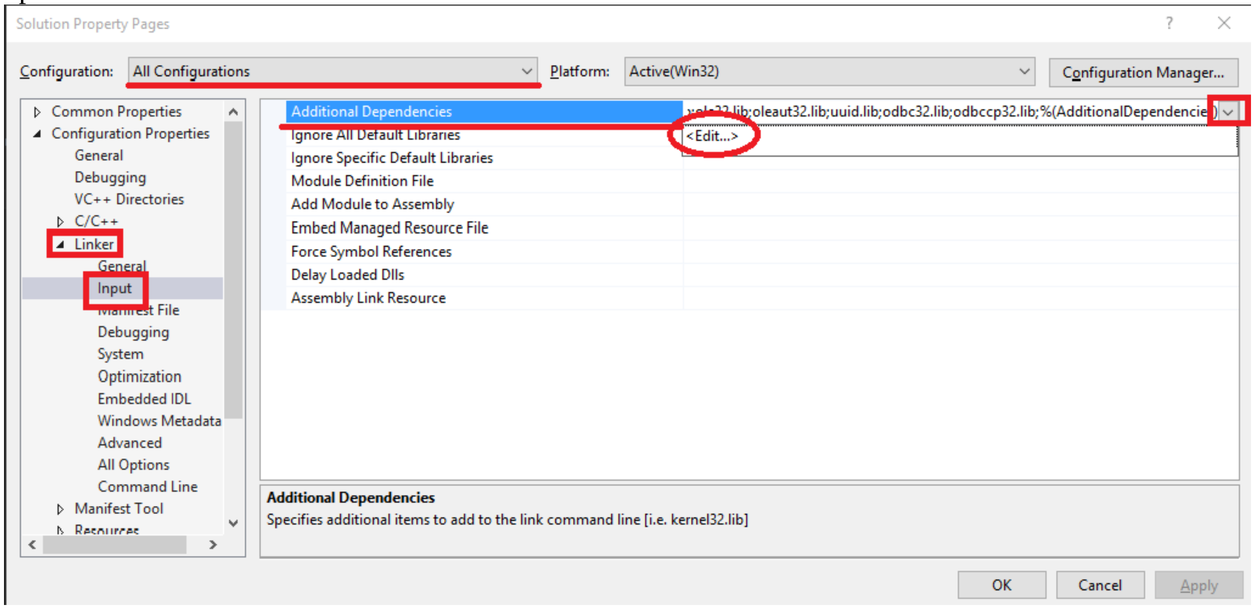


General

Executable Directories	**$(SolutionDir)include\GL\bin;$(ExecutablePath)**
Include Directories	**include;$(IncludePath)**
Reference Directories	$(VC_ReferencesPath_x86);
Library Directories	**$(SolutionDir)include\GL\lib;$(SolutionDir)include\GLFW\lib;$(LibraryPath)**
Library WinRT Directories	$(WindowsSDK_MetadataPath);
Source Directories	$(VC_SourcePath);
Exclude Directories	$(VC_IncludePath);$(WindowsSDK_IncludePath);$(MSBuild_ExecutablePath);$(VC_Libra

Library Directories
Path to use when searching for library files while building a VC++ project. Corresponds to environment variable LIB.

30) Under "Additional Dependencies" in Linker Input click on the arrow so you have access to the edit options.

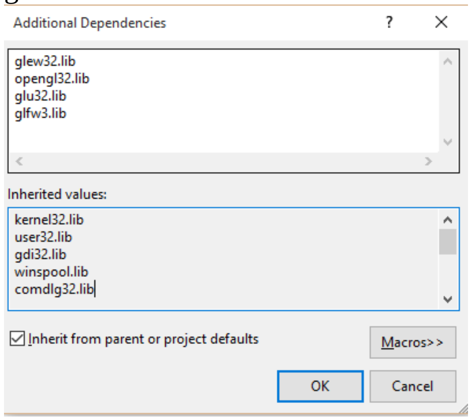


Additional Dependencies	...ole32.lib;oleaut32.lib;uuid.lib;odbc32.lib;odbccp32.lib;%(AdditionalDependencies)
Ignore All Default Libraries	<Edit...>
Ignore Specific Default Libraries	
Module Definition File	
Add Module to Assembly	
Embed Managed Resource File	
Force Symbol References	
Delay Loaded Dlls	
Assembly Link Resource	

Additional Dependencies
Specifies additional items to add to the link command line [i.e. kernel32.lib]

31) Insert the following dependencies:

glew32.lib
opengl32.lib
glu32.lib
glfw3.lib



```
glew32.lib
opengl32.lib
glu32.lib
glfw3.lib
```

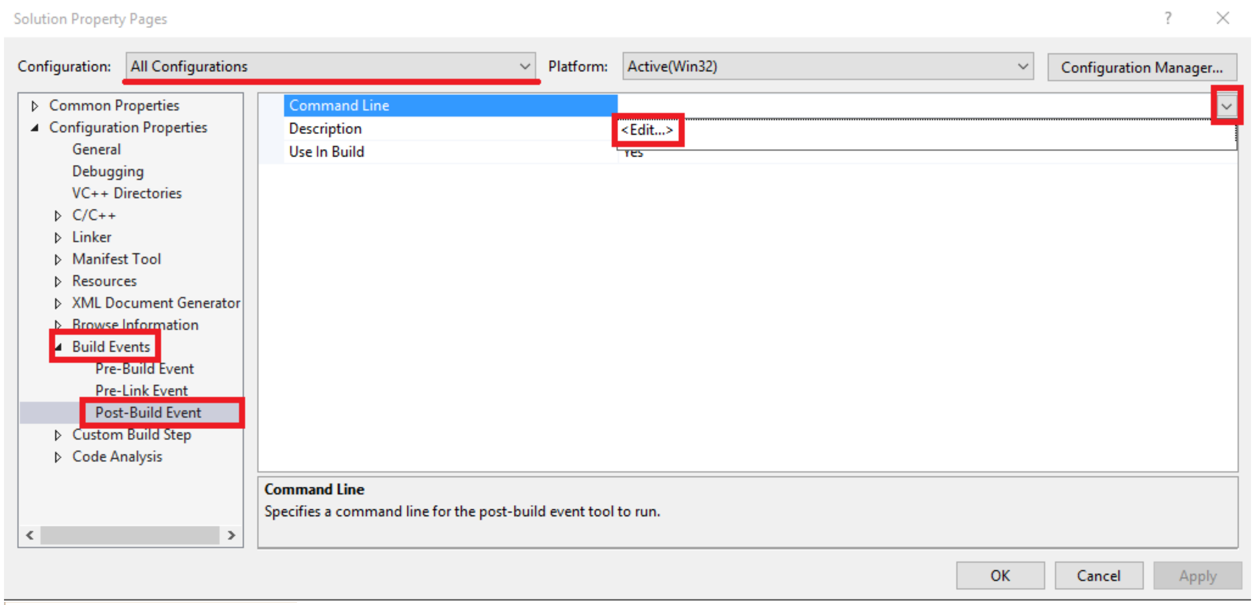
Inherited values:

```
kernel32.lib
user32.lib
gdi32.lib
winspool.lib
comdlg32.lib
```

☑ Inherit from parent or project defaults

32) In Post-Build Event under Build Events expand the arrow under Command Line and add the following ***<u>(Don't forget to change all the quotes to the right type)</u>***:

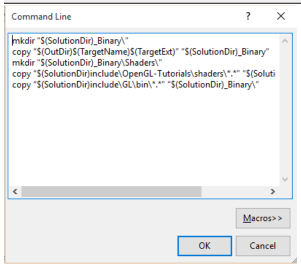
```
mkdir "$(SolutionDir)_Binary\"
copy "$(OutDir)$(TargetName)$(TargetExt)" "$(SolutionDir)_Binary"
mkdir "$(SolutionDir)_Binary\Shaders\"
copy "$(SolutionDir)include\OpenGL-Tutorials\shaders\*.*" "$(SolutionDir)_Binary\Shaders\"
copy "$(SolutionDir)include\GL\bin\*.*" "$(SolutionDir)_Binary\"
```



Configuration: All Configurations
Platform: Active(Win32)

Command Line	
Description	
Use In Build	Yes

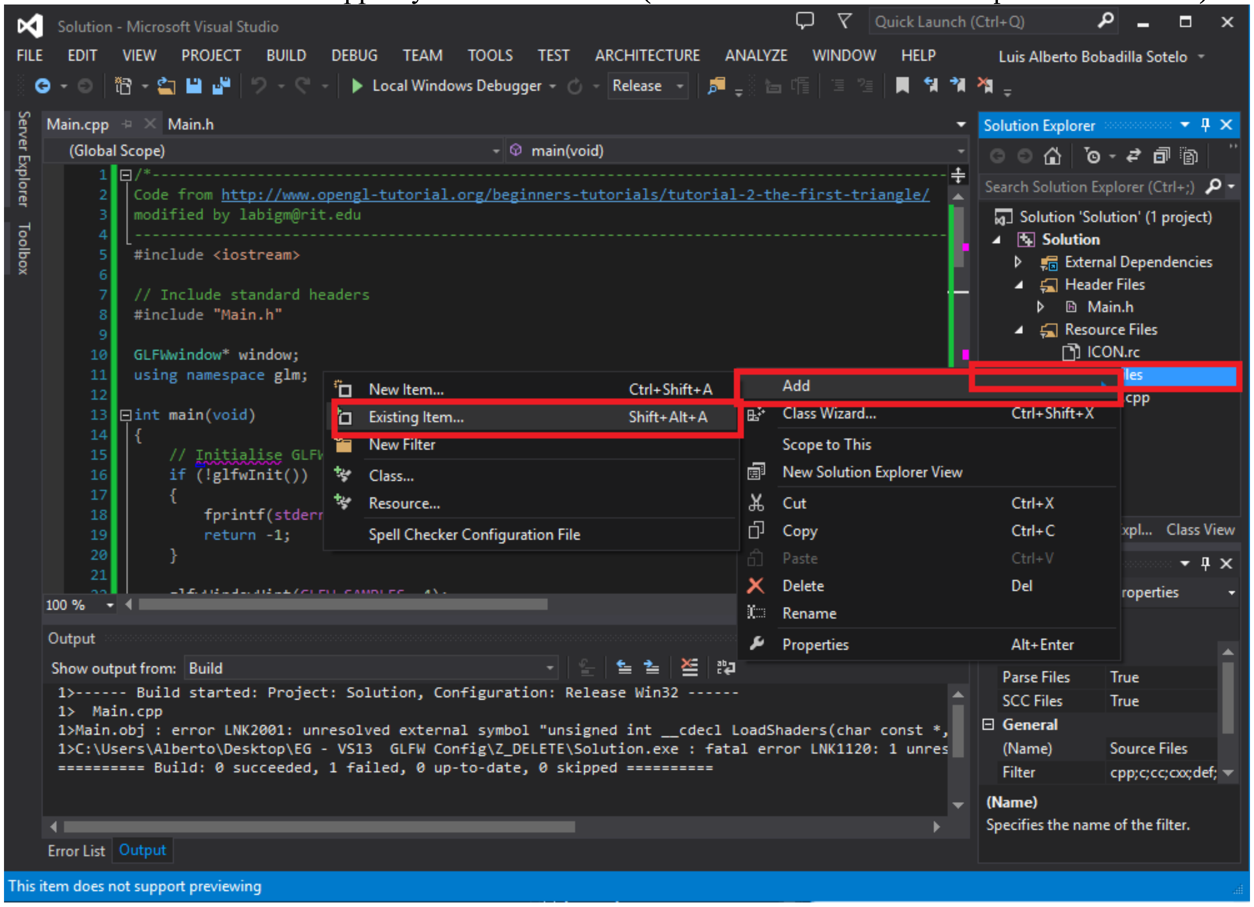
Command Line
Specifies a command line for the post-build event tool to run.



```
mkdir "$(SolutionDir)_Binary\"
copy "$(OutDir)$(TargetName)$(TargetExt)" "$(SolutionDir)_Binary"
mkdir "$(SolutionDir)_Binary\Shaders\"
copy "$(SolutionDir)include\OpenGL-Tutorials\shaders\*.*" "$(Soluti
copy "$(SolutionDir)include\GL\bin\*.*" "$(SolutionDir)_Binary\"
```

Remember that usually text editors change the "quotes" for text quotes (opening and closing) and they are different symbols than the ones using in most other software be sure that if you are copy-pasting code you replace the quotes by hand or you will have an error.

33) Replace your main.cpp file's with the one provided and include the files shader.cpp to your "source files" folder and the shader.hpp to your "header files" (the files live under include\OpenGL-Tutorials\):

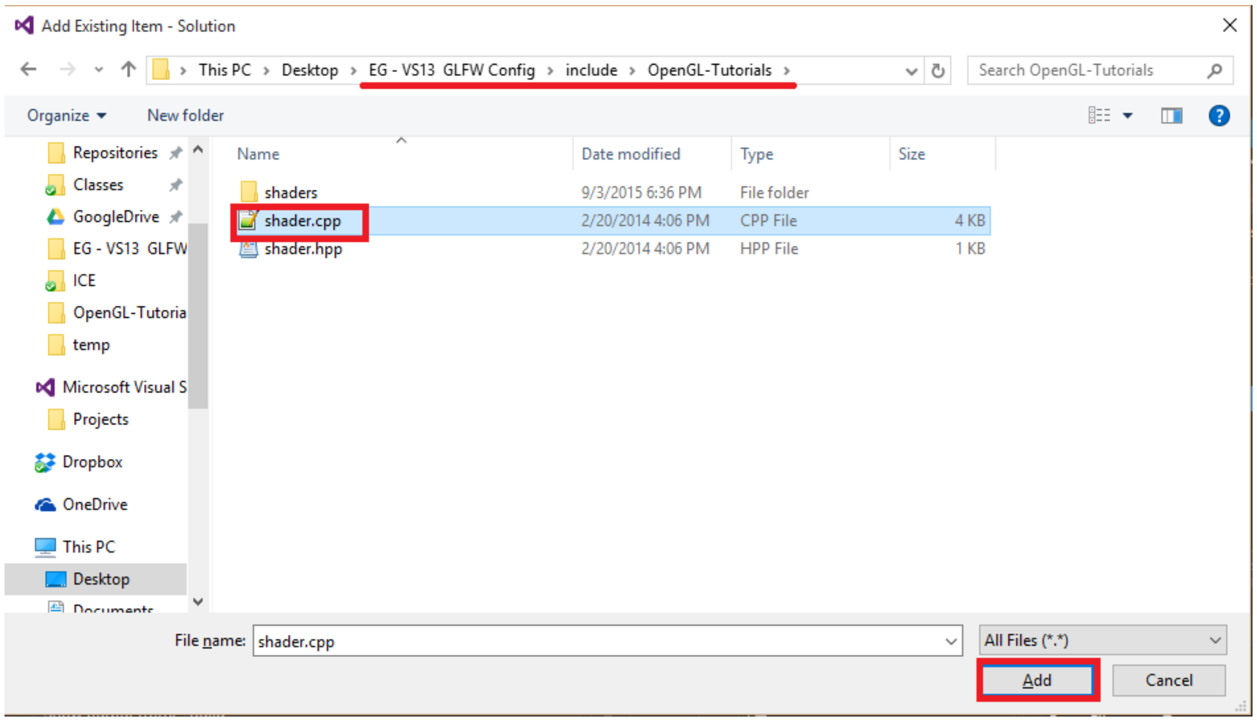


```
/*------------------------------------------------------------------------------
Code from http://www.opengl-tutorial.org/beginners-tutorials/tutorial-2-the-first-triangle/
modified by labigm@rit.edu
--------------------------------------------------------------------------------
#include <iostream>

// Include standard headers
#include "Main.h"

GLFWwindow* window;
using namespace glm;

int main(void)
{
    // Initialise GLFW
    if (!glfwInit())
    {
        fprintf(stderr
        return -1;
    }
```

```
1>------ Build started: Project: Solution, Configuration: Release Win32 ------
1>  Main.cpp
1>Main.obj : error LNK2001: unresolved external symbol "unsigned int __cdecl LoadShaders(char const *,
1>C:\Users\Alberto\Desktop\EG - VS13  GLFW Config\Z_DELETE\Solution.exe : fatal error LNK1120: 1 unres
========== Build: 0 succeeded, 1 failed, 0 up-to-date, 0 skipped ==========
```

Parse Files	True
SCC Files	True
General	
(Name)	Source Files
Filter	cpp;c;cc;cxx;def;

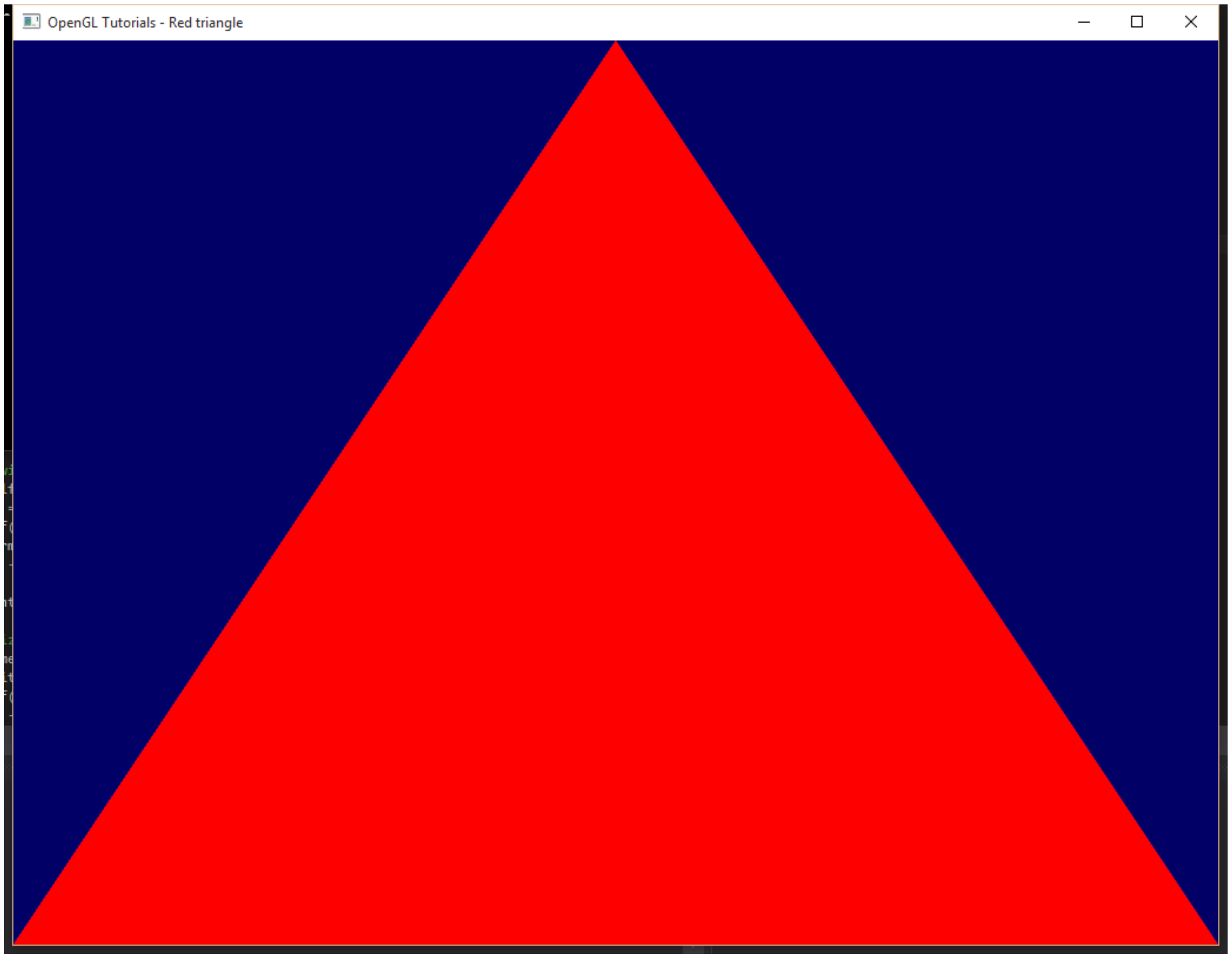
(Name)
Specifies the name of the filter.



Name	Date modified	Type	Size
shaders	9/3/2015 6:36 PM	File folder	
shader.cpp	2/20/2014 4:06 PM	CPP File	4 KB
shader.hpp	2/20/2014 4:06 PM	HPP File	1 KB

File name: shader.cpp
All Files (*.*)

34) Should you have no errors, the window will display:



35) Remove the sdf file; the hidden .vs folder and the intermediary Z_Delete folder and zip your folder.

36) Zip your project and show it to the TA or Teacher, you should also show them the _Binary folder and execute the release version of the file (the one without the _d appended)

NOTE: This configuration will only work for an x86 solution, you can apply the same changes to a 64 solution but the corresponding libs and dlls are needed (I only provided 86 versions)