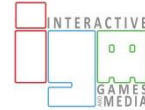**Rochester Institute of Technology**
Golisano College of Computing and Information
Sciences
School of Interactive Games and Media
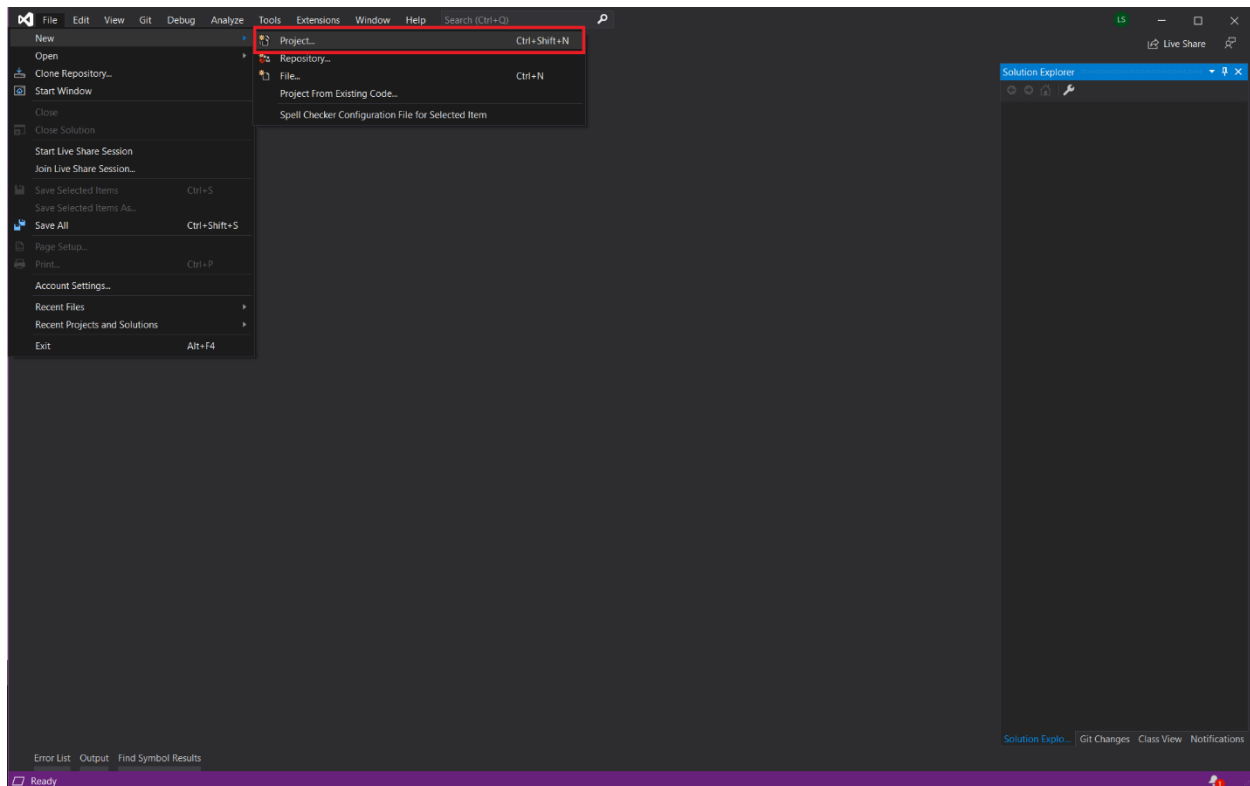**2145 Golisano Hall – (585) 475-7680**

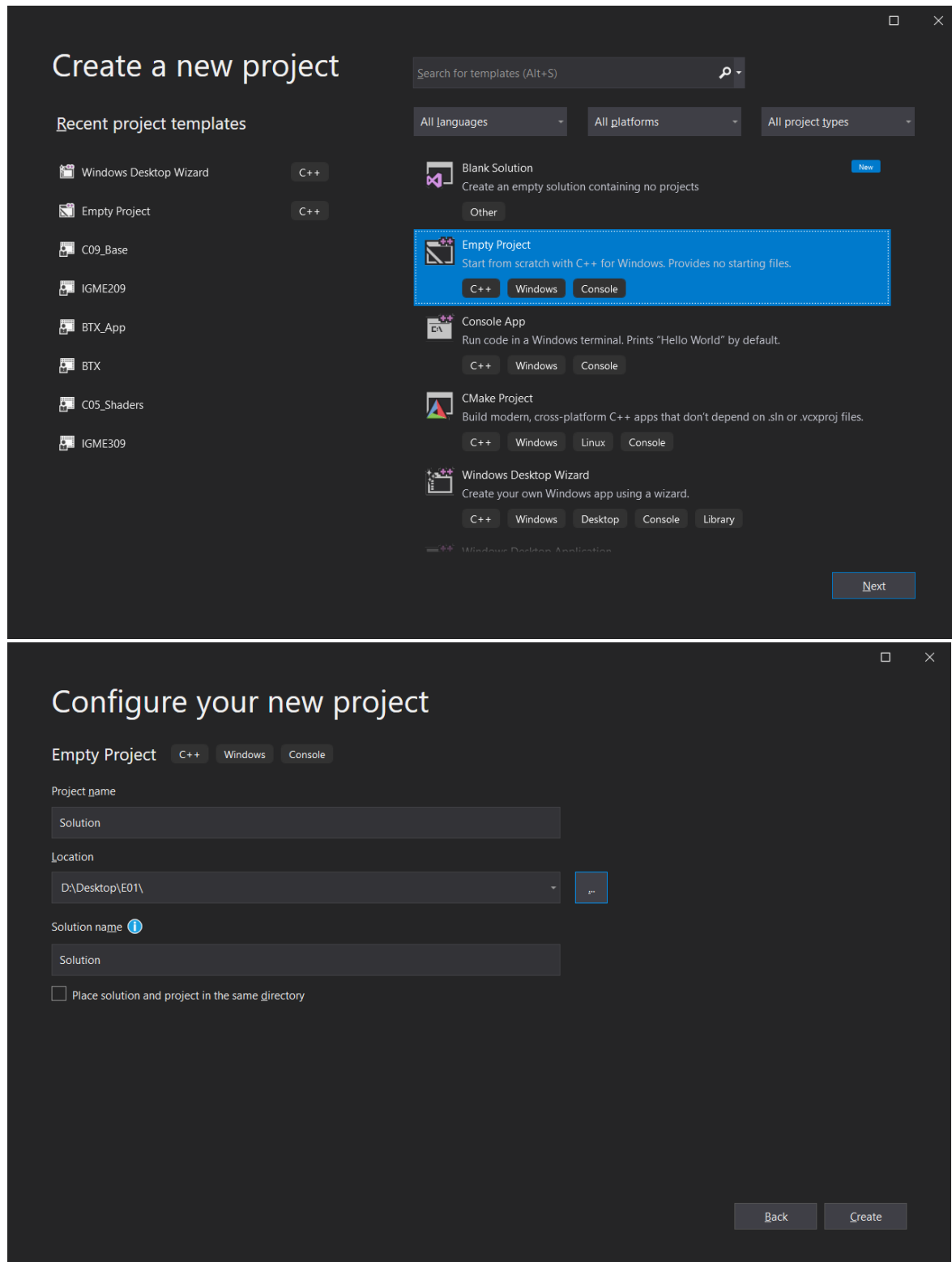# Data Structures & Algorithms for Games & Simulation II
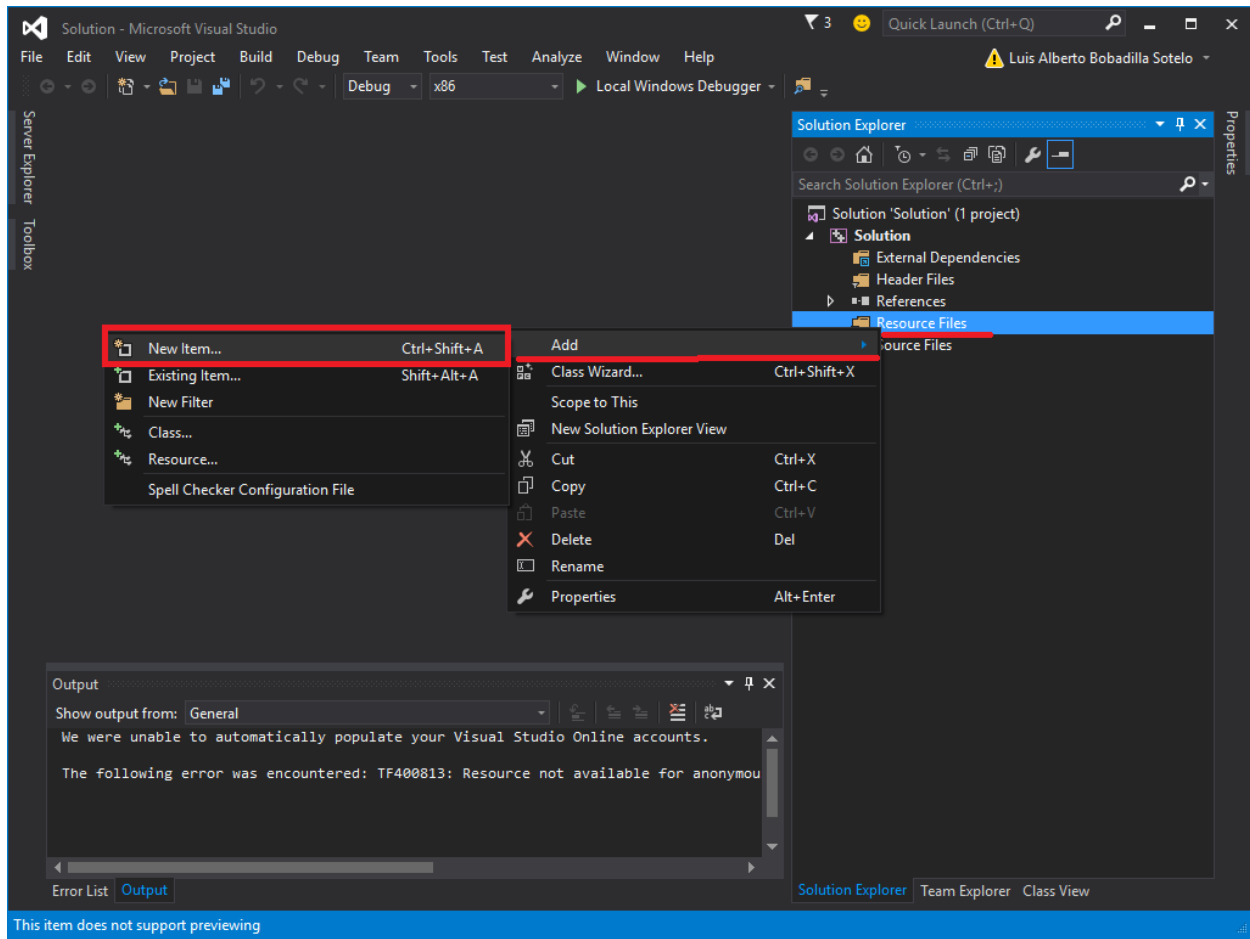# IGME 309
E01: Visual Studio Configuration

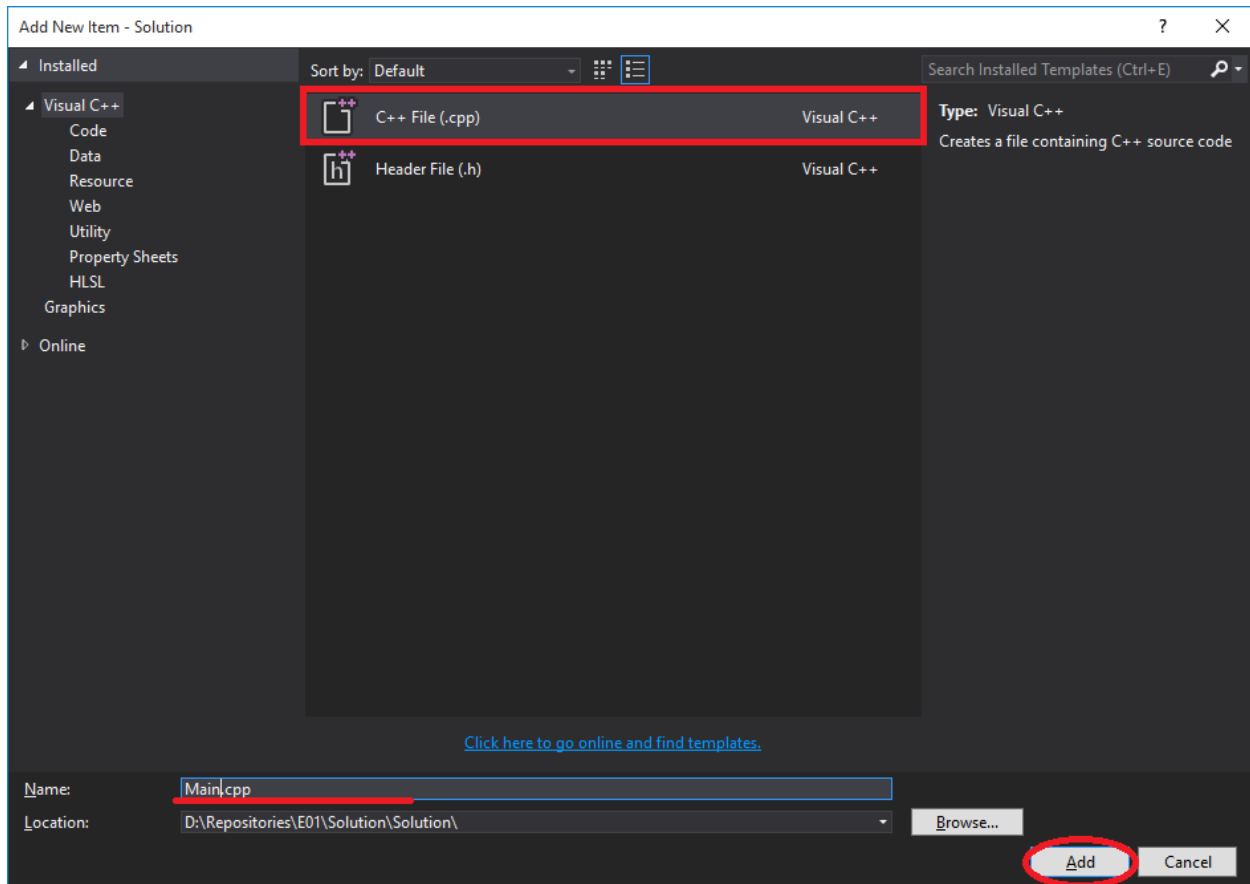1) Open a New Visual Studio Window and create a new Project (FILE/New/Project…)

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

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

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

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)

```cpp
/*--------------------------------------------------------------------------------------
------------
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;
}
```
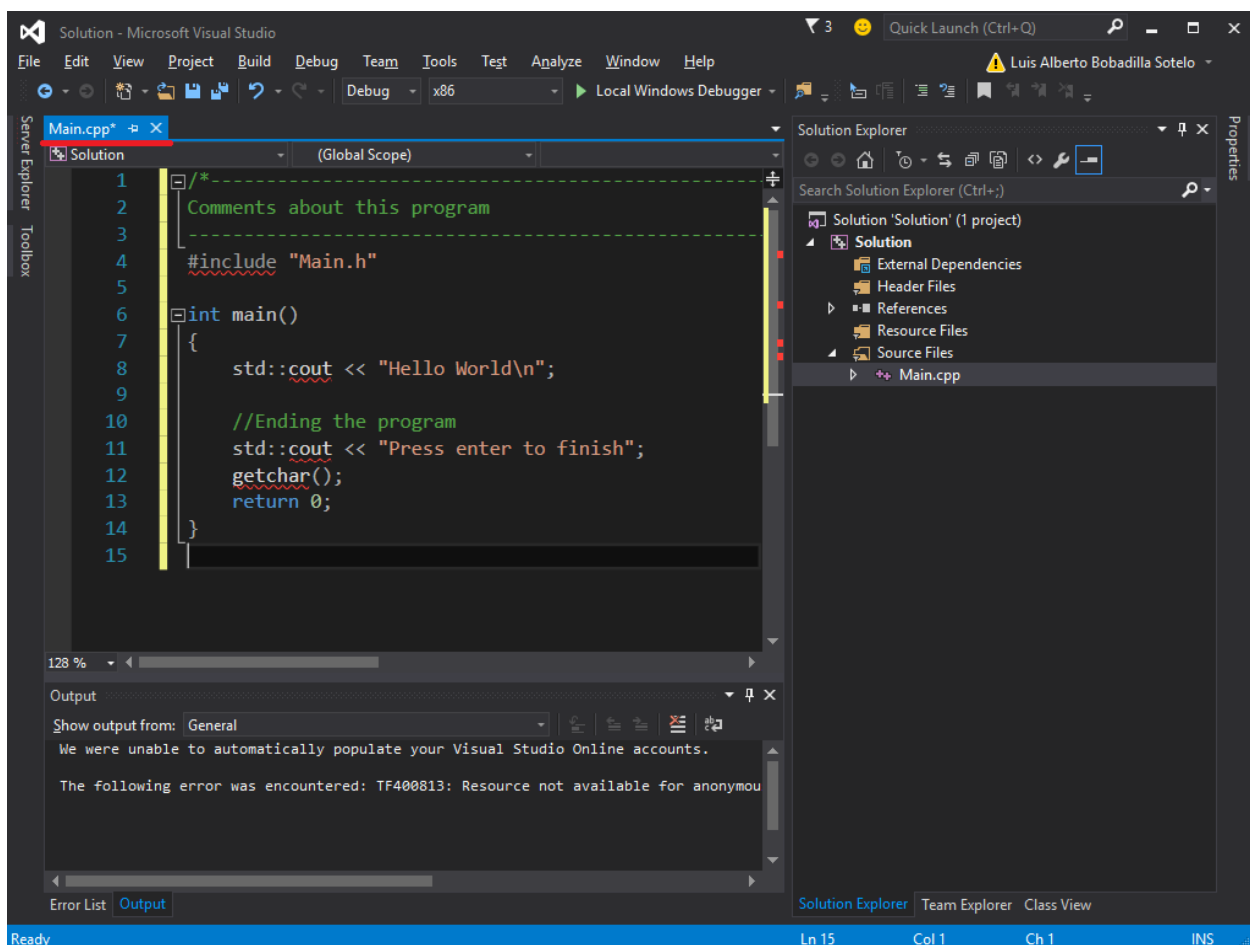

Also remember that if you copy-paste code you need to replace the "quotes" to actual " " (they have a different ASCII value and word replaces " for the curved ones for easy reading.

6) Add a new header file IN THE HEADERS FOLDER (Header Files/Add/New Item…)
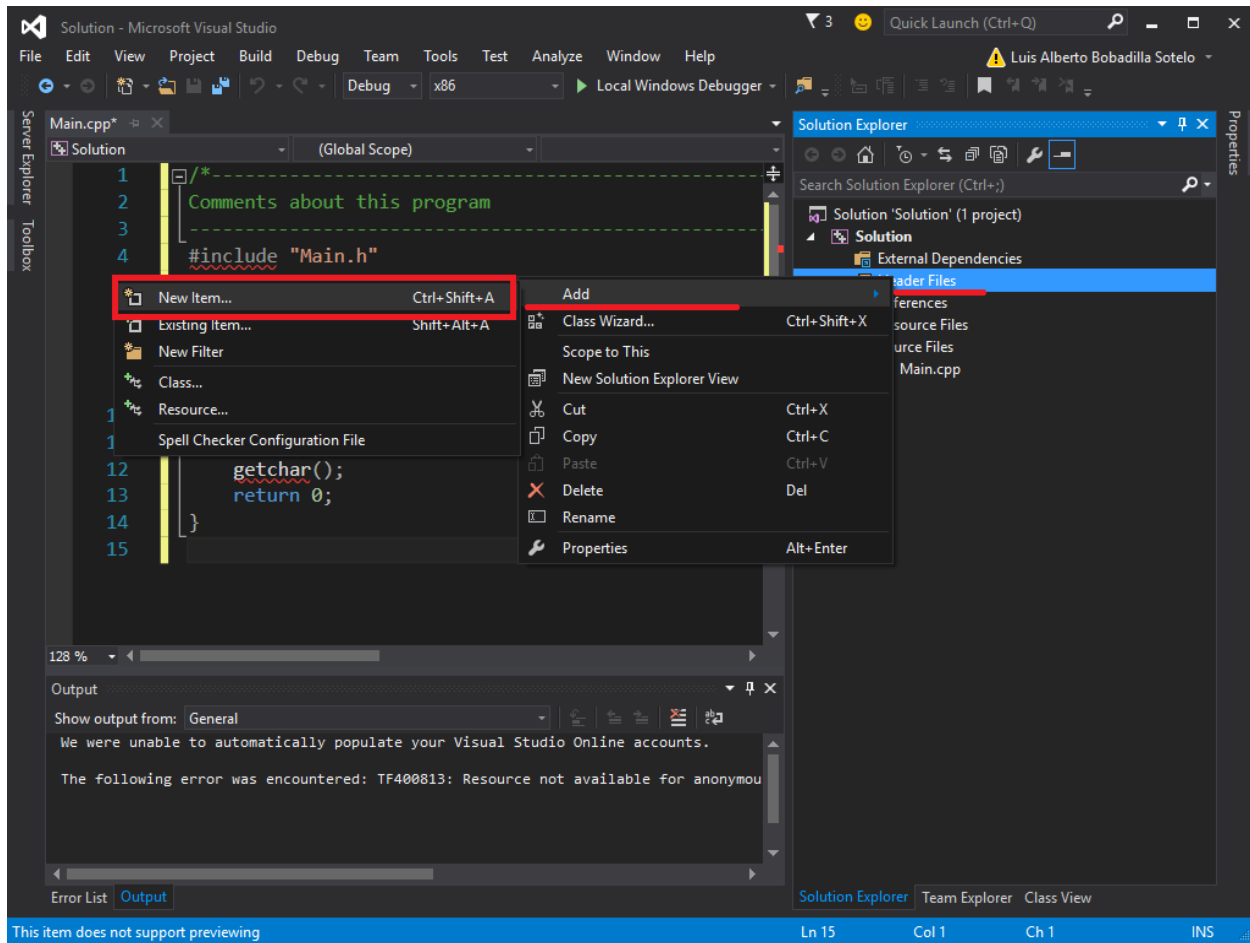
7) Name the file however you want but remember to include the header from the cpp file
(Name it "Main" for the sake of this example) and hit OK
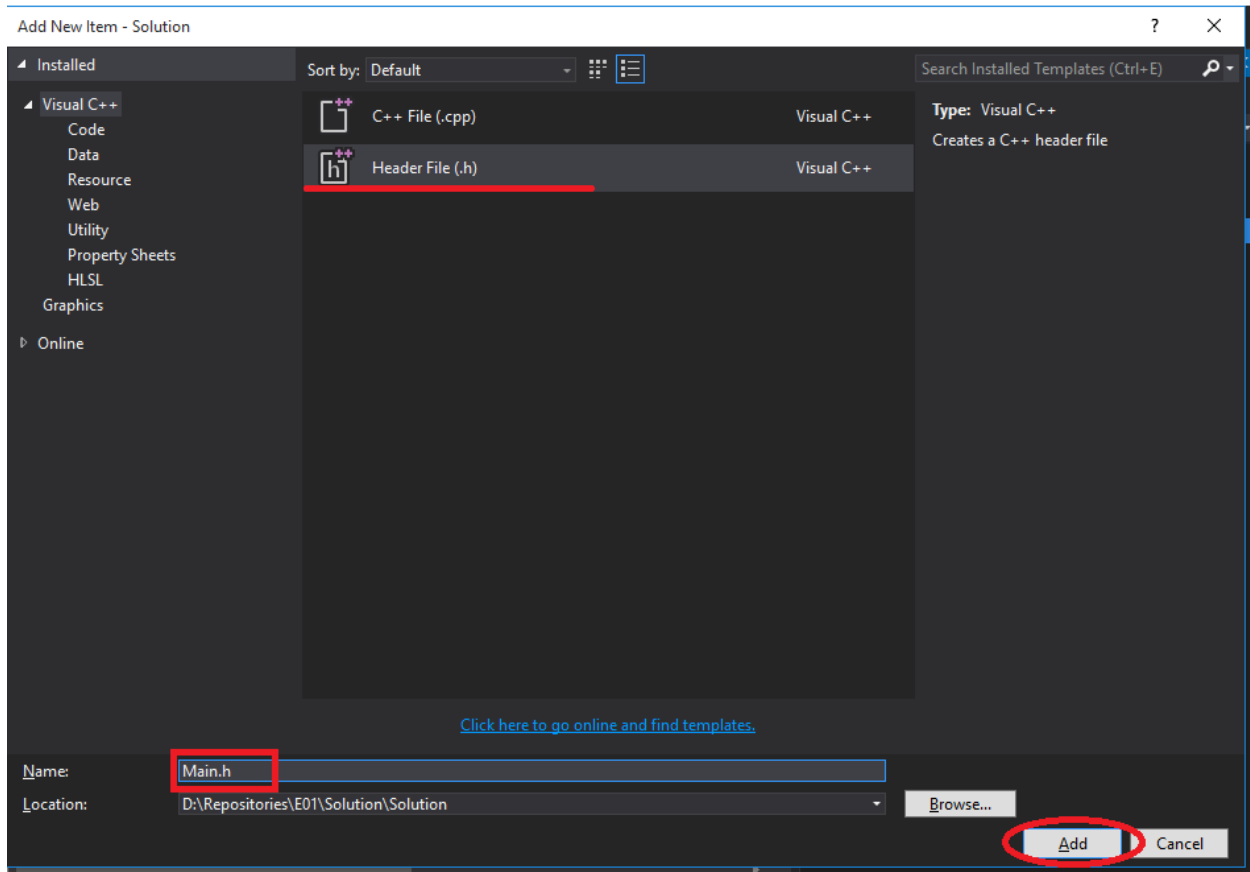
8) Create the protectors for your file, so the compiler don't try to compile the file more than once. And include the headers that you will use in your project. (for this example just copy paste these lines,)

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

#include <iostream>

#endif //_MAIN_H
```
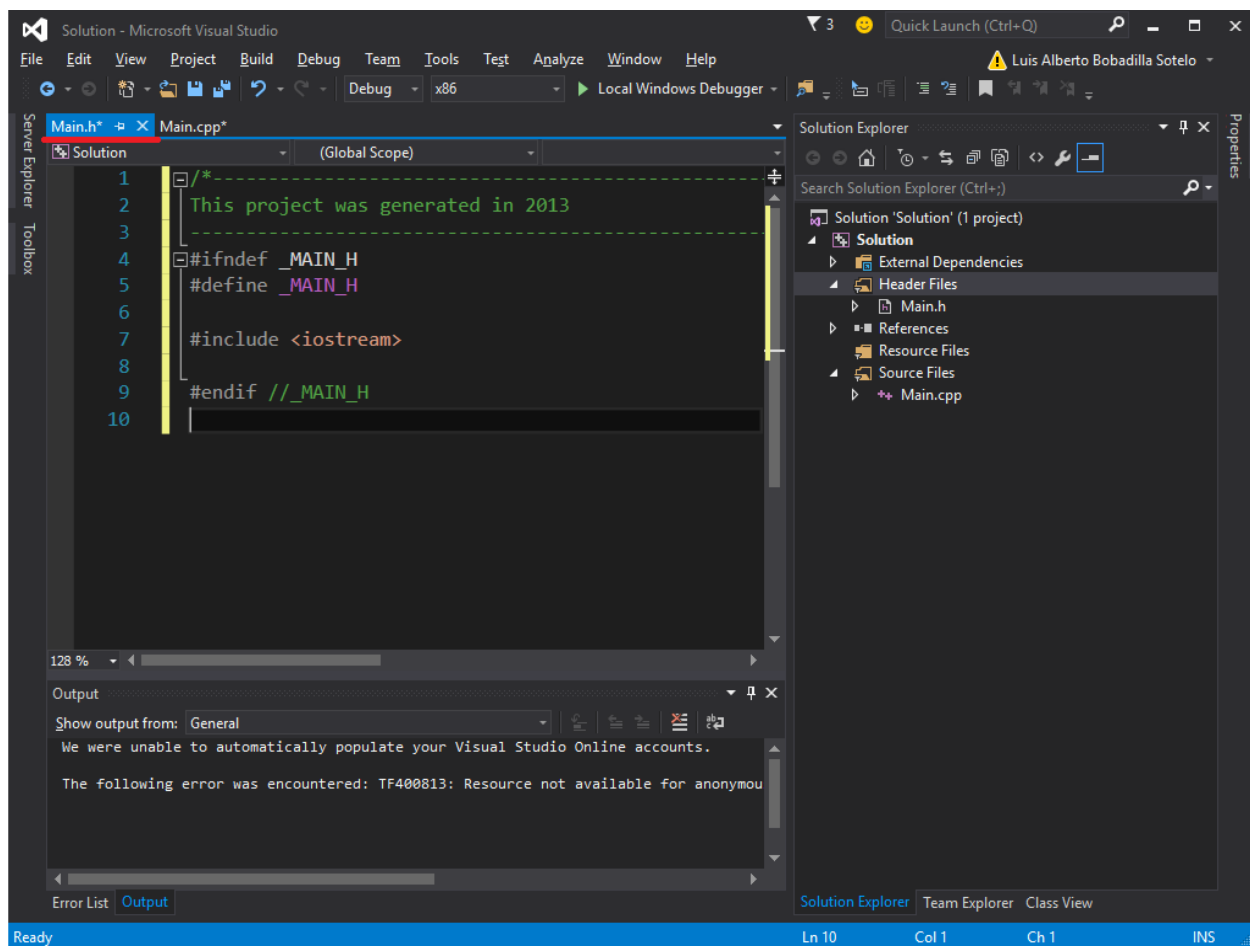

The difference with the #ifndef/#define/#endif and #pragma once is that #pragma is non-standard and it would just work right in most of the compilers but not in all, causing differences in the semantic, while #ifndef/#define/#endif is standard working in all compilers and its usually faster. If you don't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

9) Go to your Solution properties. (PROJECT/[Solution] Properties…)
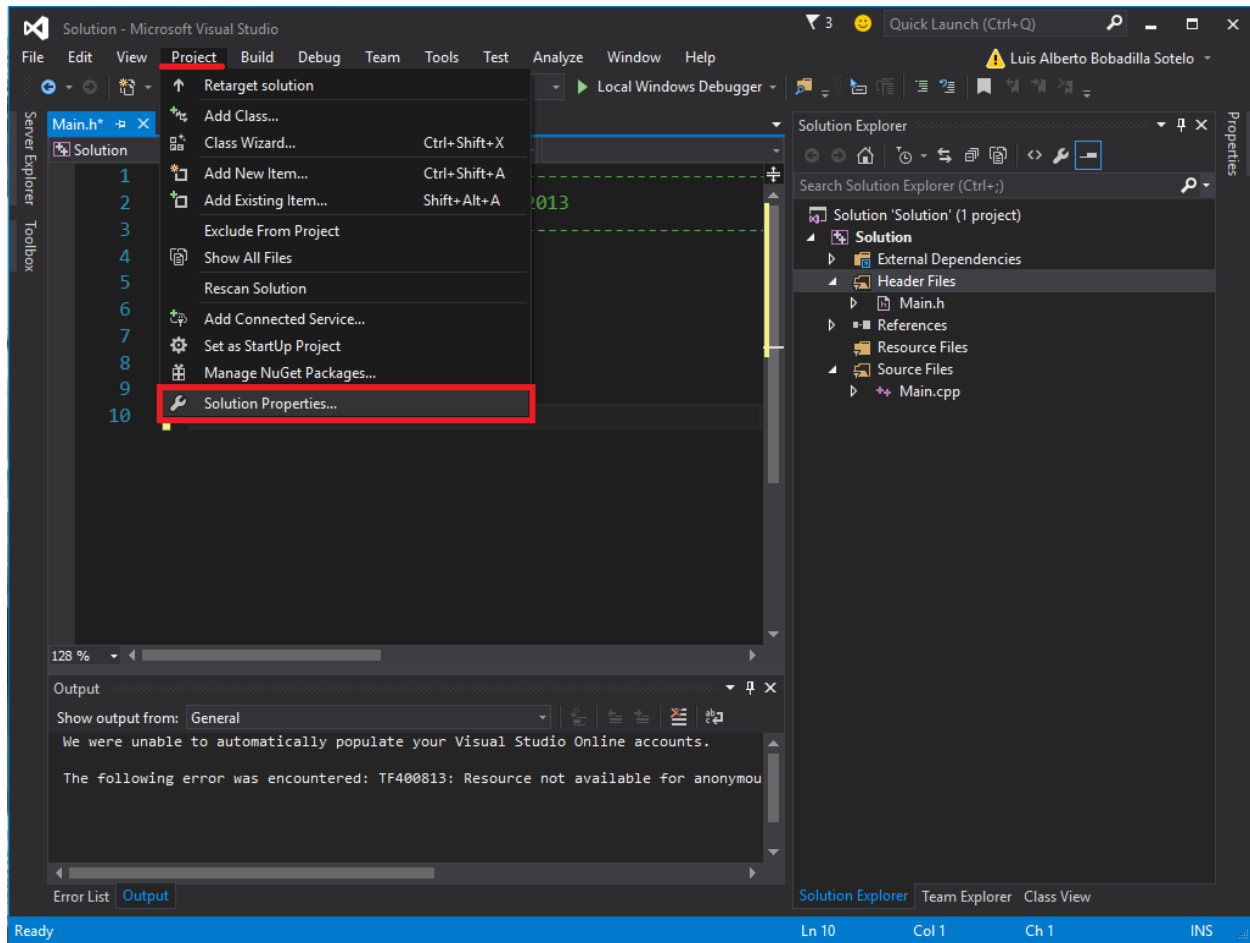
10) Click on Configuration Properties and make sure you are changing the settings for **all the configurations and Win32 platform** go to General and change the Output Directory, the Intermediate Directory and the Target Name fields like follows and apply your changes:
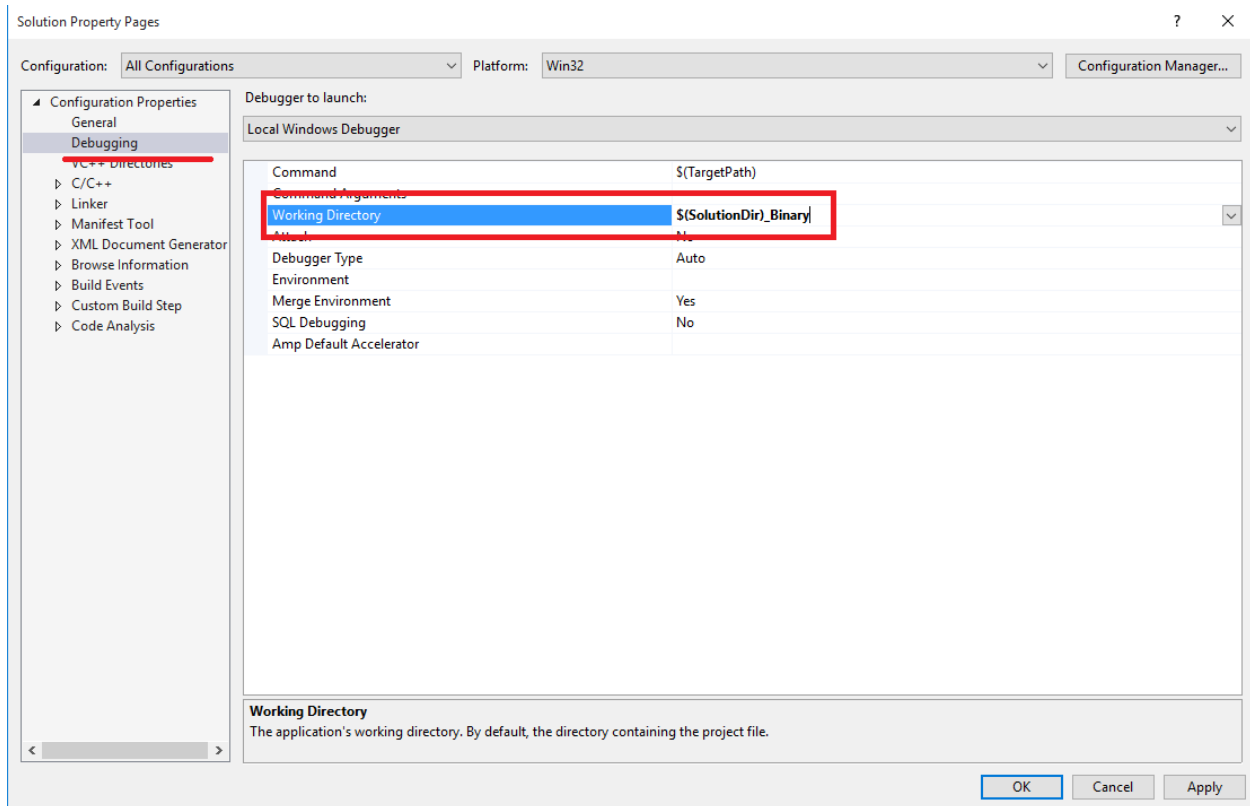
Output Directory: $(SolutionDir)Z_DELETE\

Intermediate Directory: $(SolutionDir)Z_DELETE\$(ProjectName)\$(Configuration)\

Target Name: $(ProjectName)
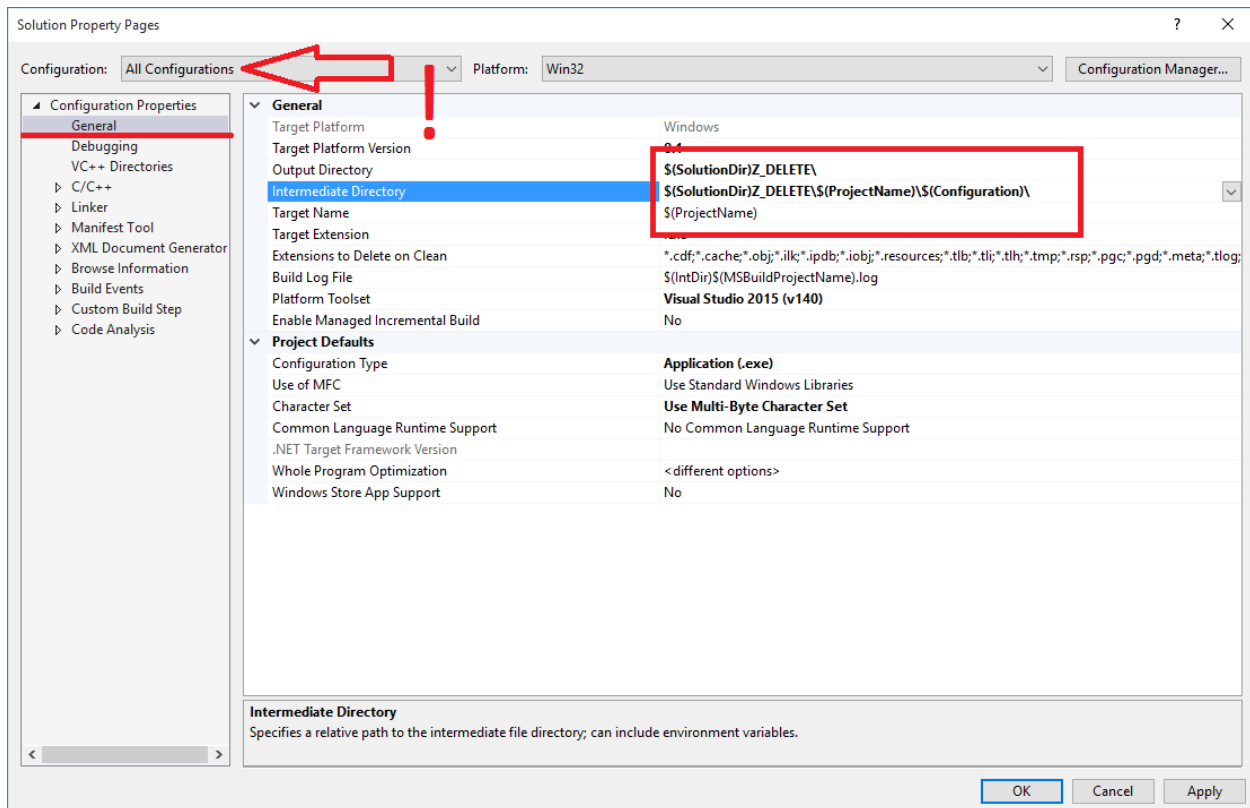
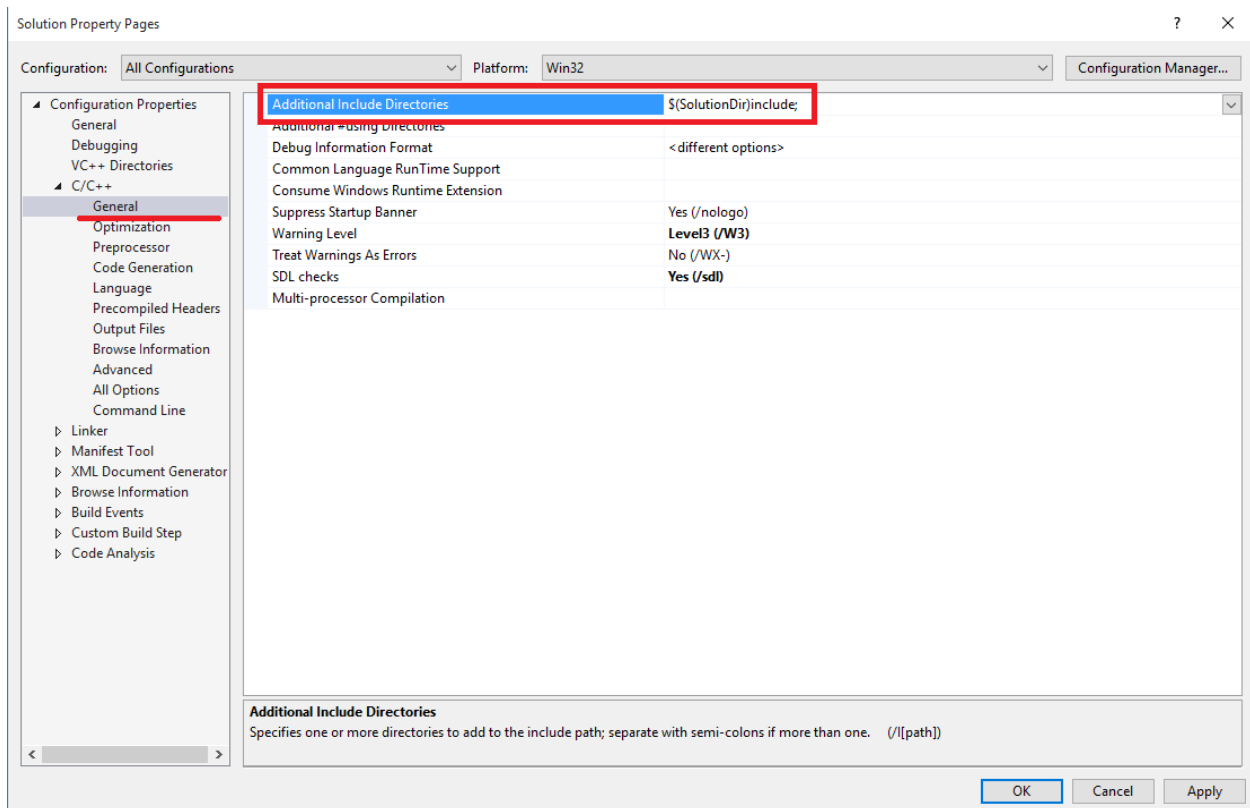
Do not forget to apply your changes!

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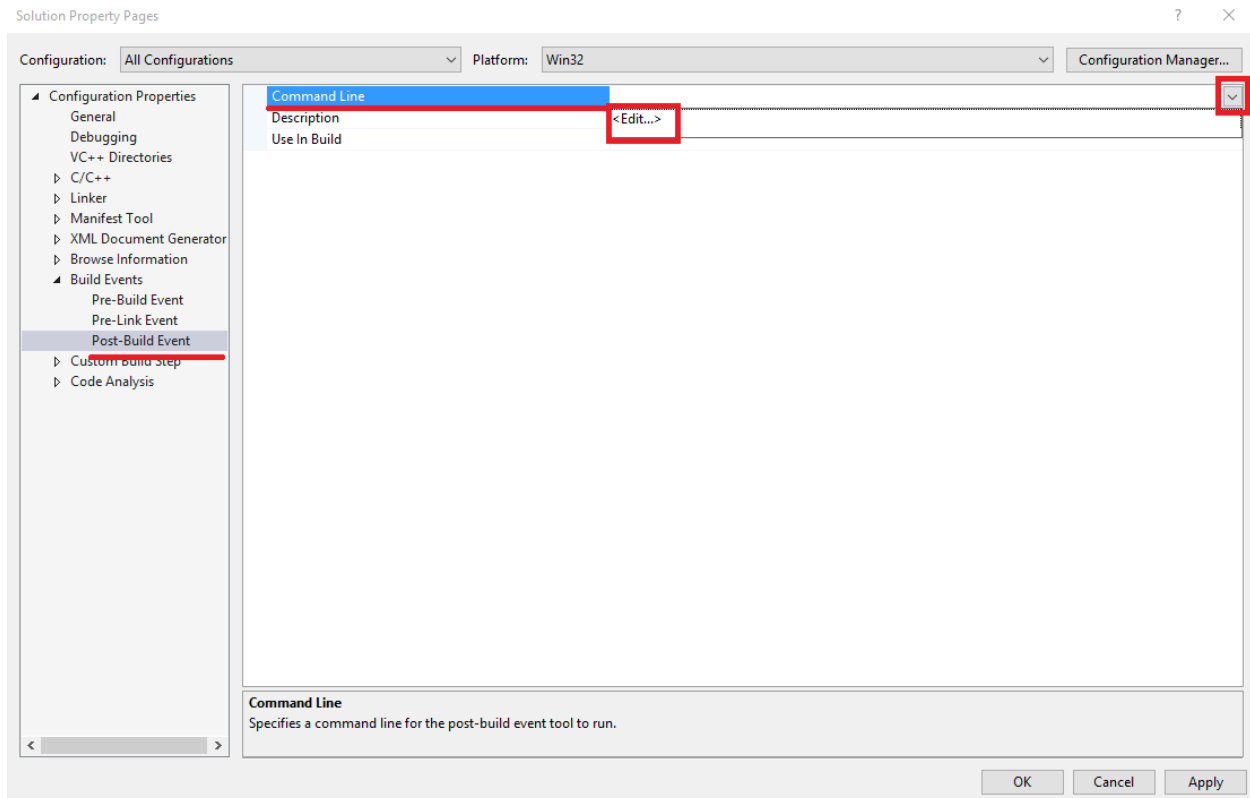$(SolutionDir)_Binary

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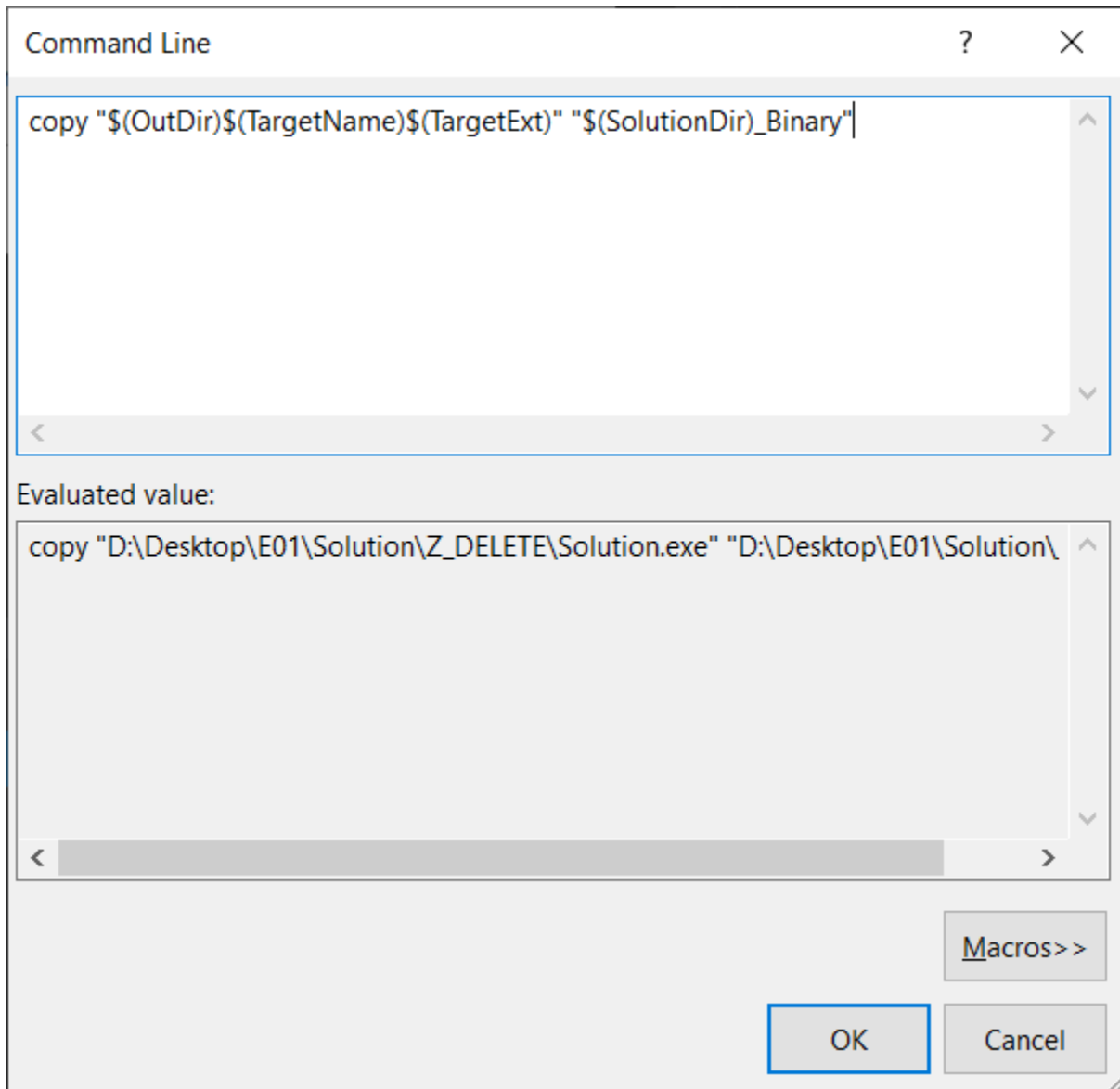$(SolutionDir)include;

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...>
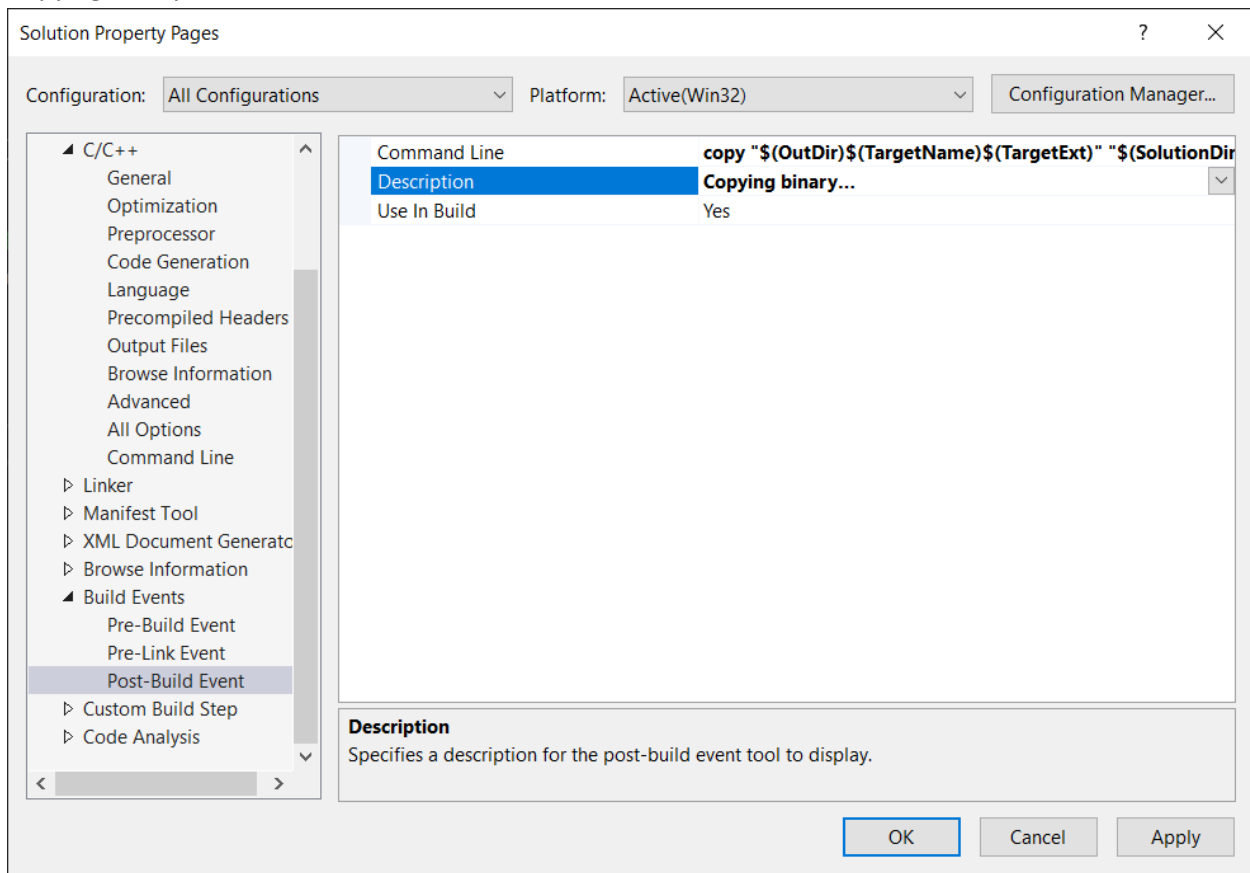
Then add (remember if you are copy-pasting, replace the " for basic double quotes, the ones you get from word are opening an closing double quotes, not the symbol you need), be extra sure that you are adding a space between your parts (3 parts 2 spaces in between):
copy "$(OutDir)$(TargetName)$(TargetExt)" "$(SolutionDir)_Binary"

Command Line   ?   ✕

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

Evaluated value:

```
copy "D:\Desktop\E01\Solution\Z_DELETE\Solution.exe" "D:\Desktop\E01\Solution\
```

Macros>>

OK   Cancel

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

<u>Copying binary…</u>

<u>Remember that usually text editors change the "quotes" for text quotes (opening and closing) and they are different symbols than the ones used in other software, be sure that if you are copy-pasting code you replace the quotes by hand or you will have an error. Also respect the spaces.</u>

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.
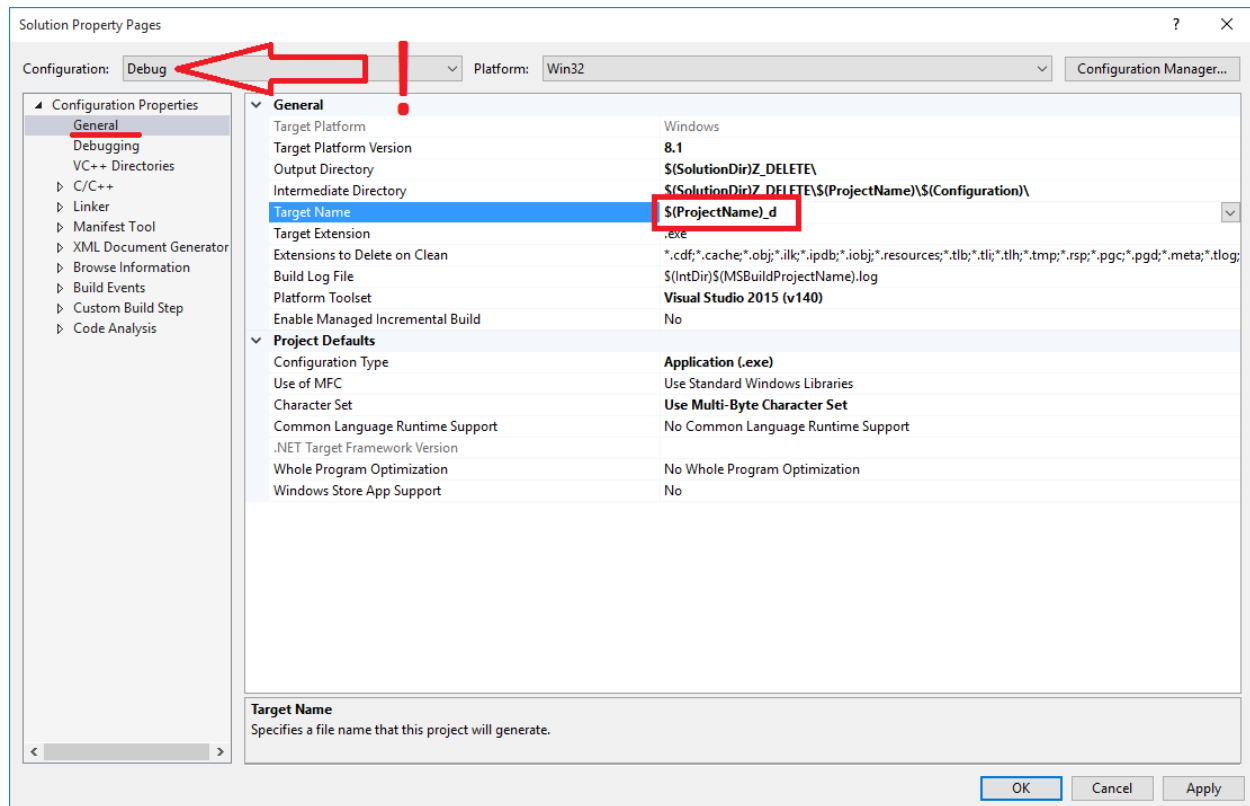
15) Still in Debug configuration in C/C++ / Preprocessor/ Preprocessor Definitions click on the arrow and add DEBUG
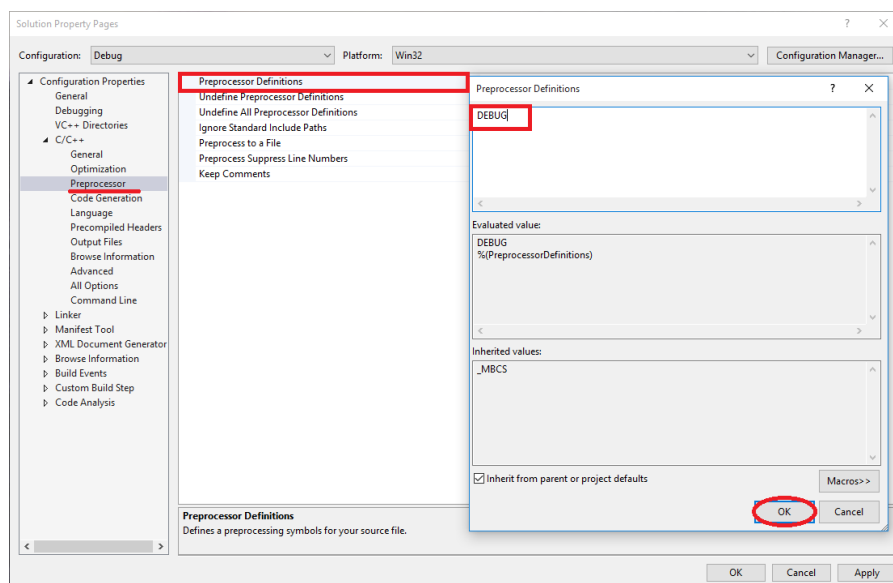
With this you are defining DEBUG as a keyword for your entire project.
With this configuration you will add all the intermediary and binary files to a folder called Z_DELETE at the same level as your solution and will copy the binary to a new folder called _Binary.

__Hit OK to close the configuration now__

16) Copy the provided res folder at the same level of your sln file and make a new _Binary folder:
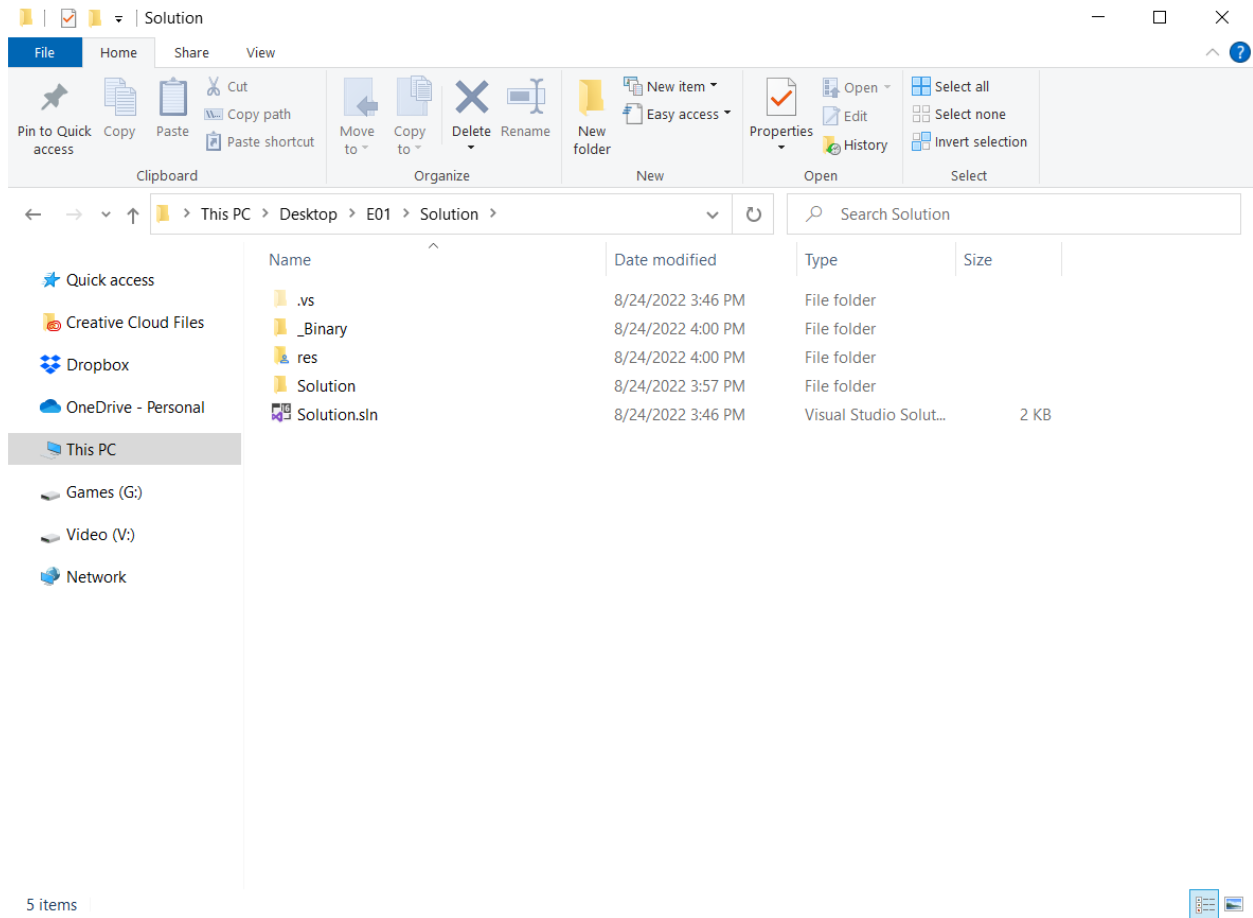
17) Add the ICON.rc into the RESOURCES FOLDER (right-click Resources Files/Add/Existing Item)
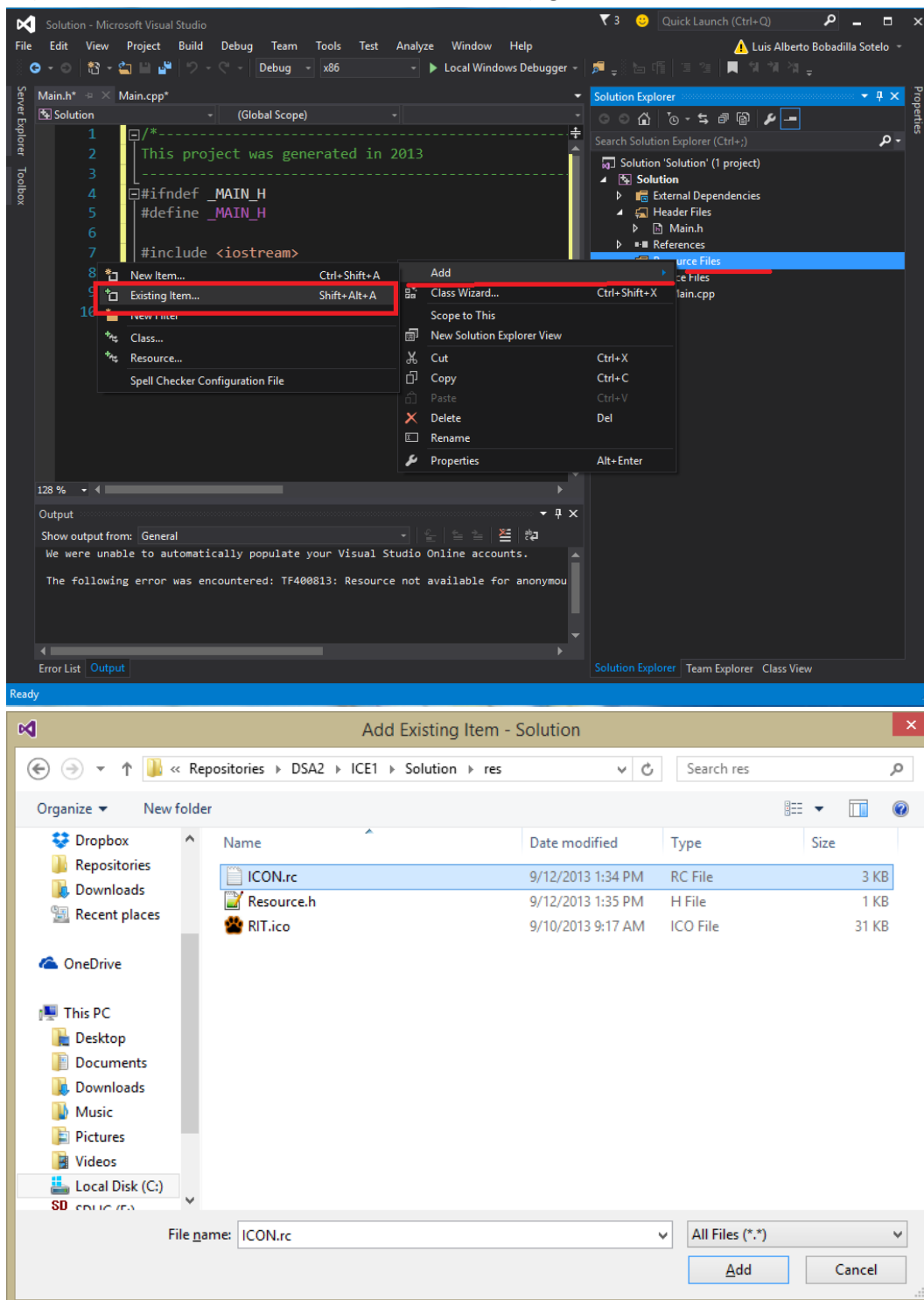

This will add an Icon to your application, if you ever want to add a different icon to your project you just need to specify said icon in the ICON.rc file accordingly (You can edit the file in Note Pad, or the icon directly in VS).

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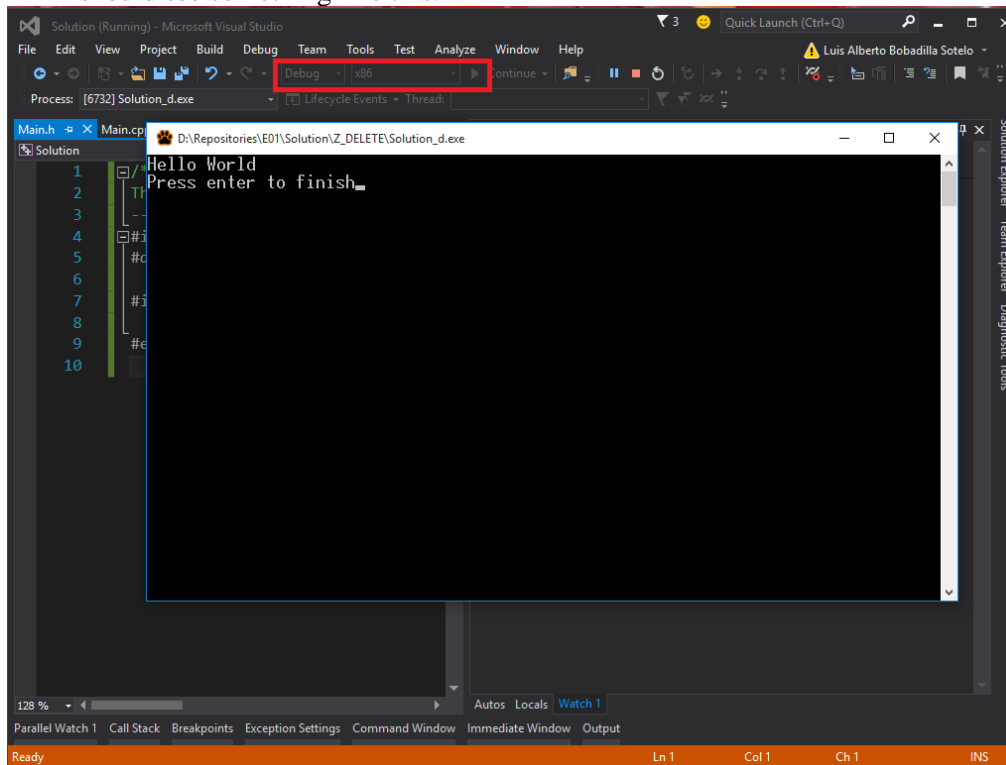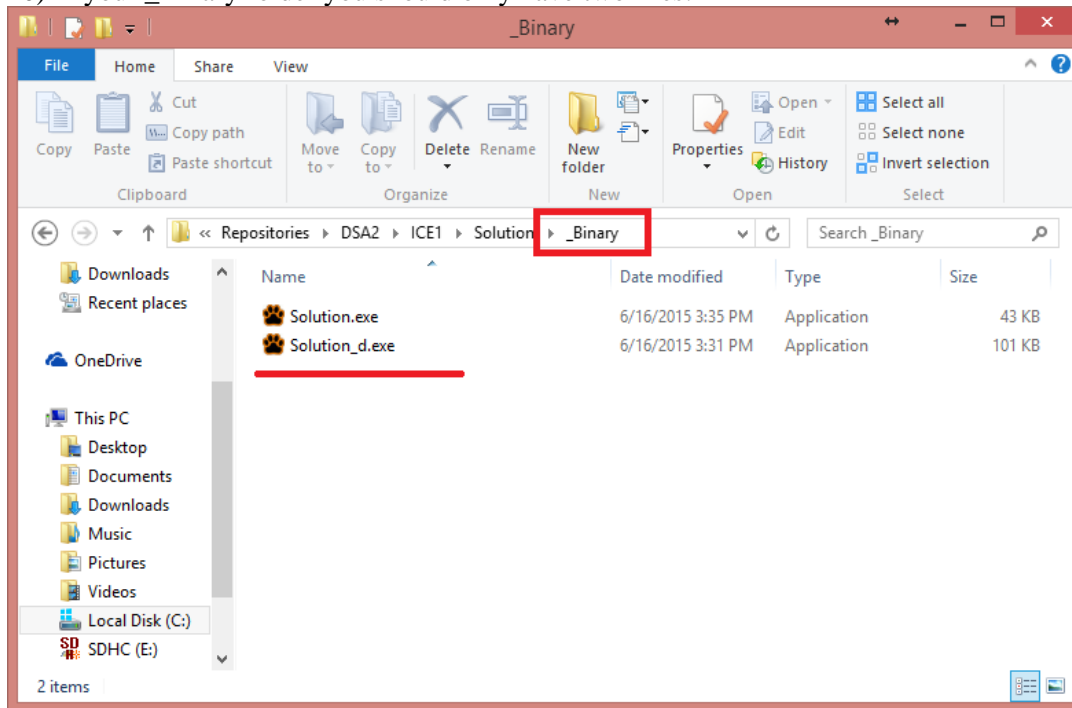

As you can see the name of the executable ends in _d

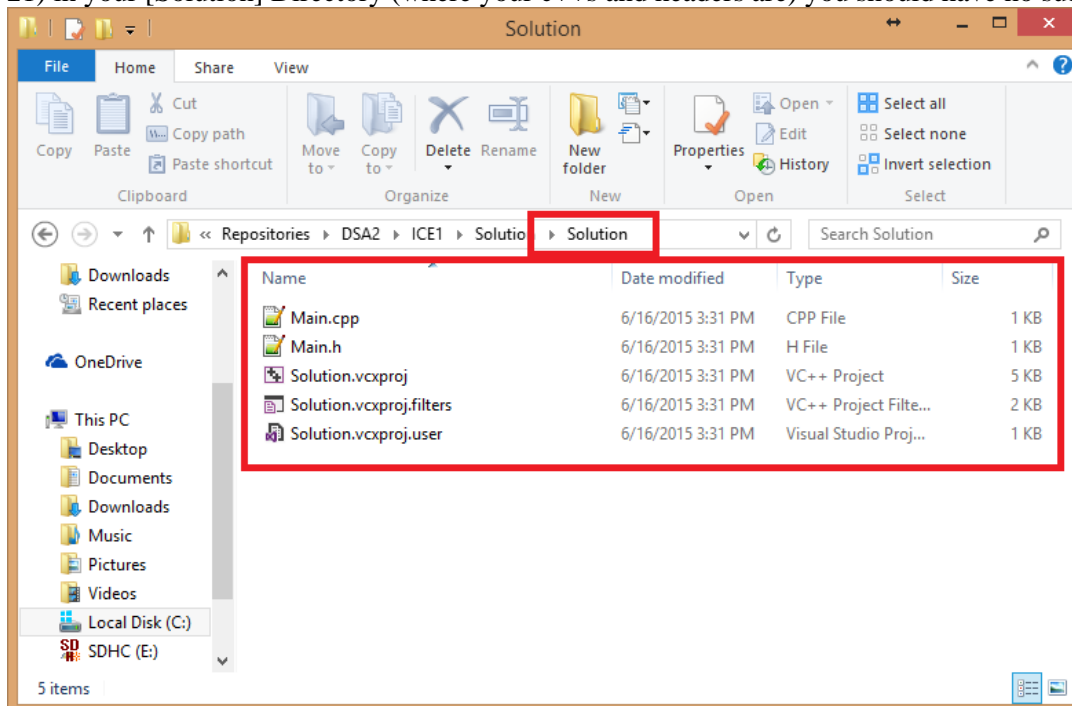19) Run your project in Release mode and you should see something like this:
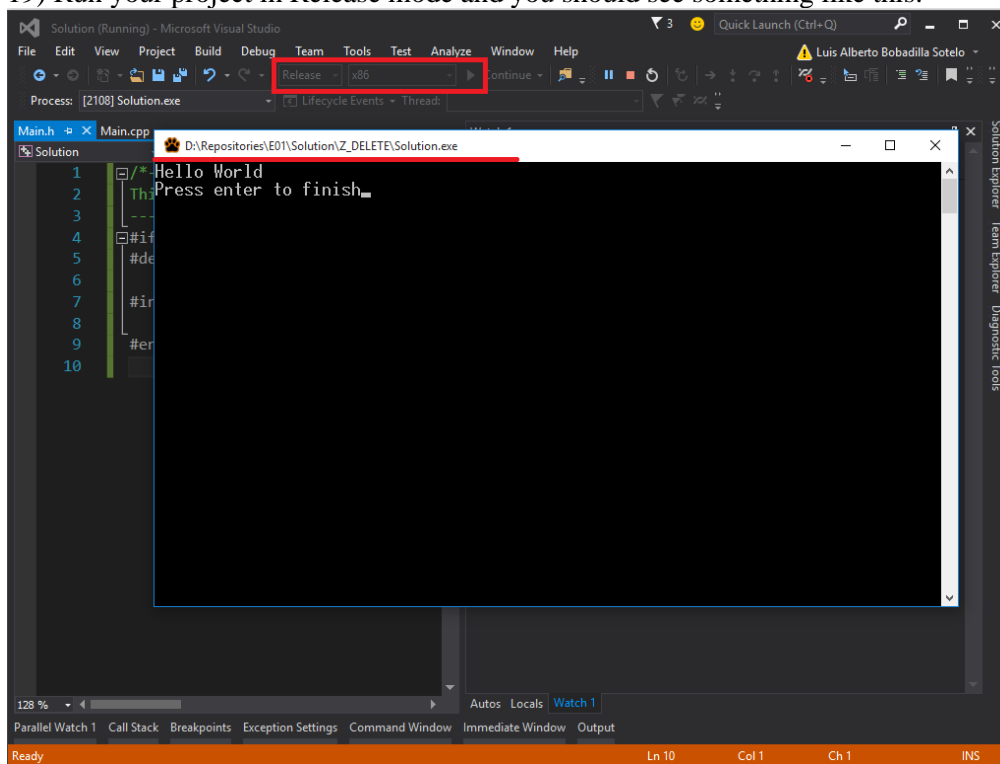
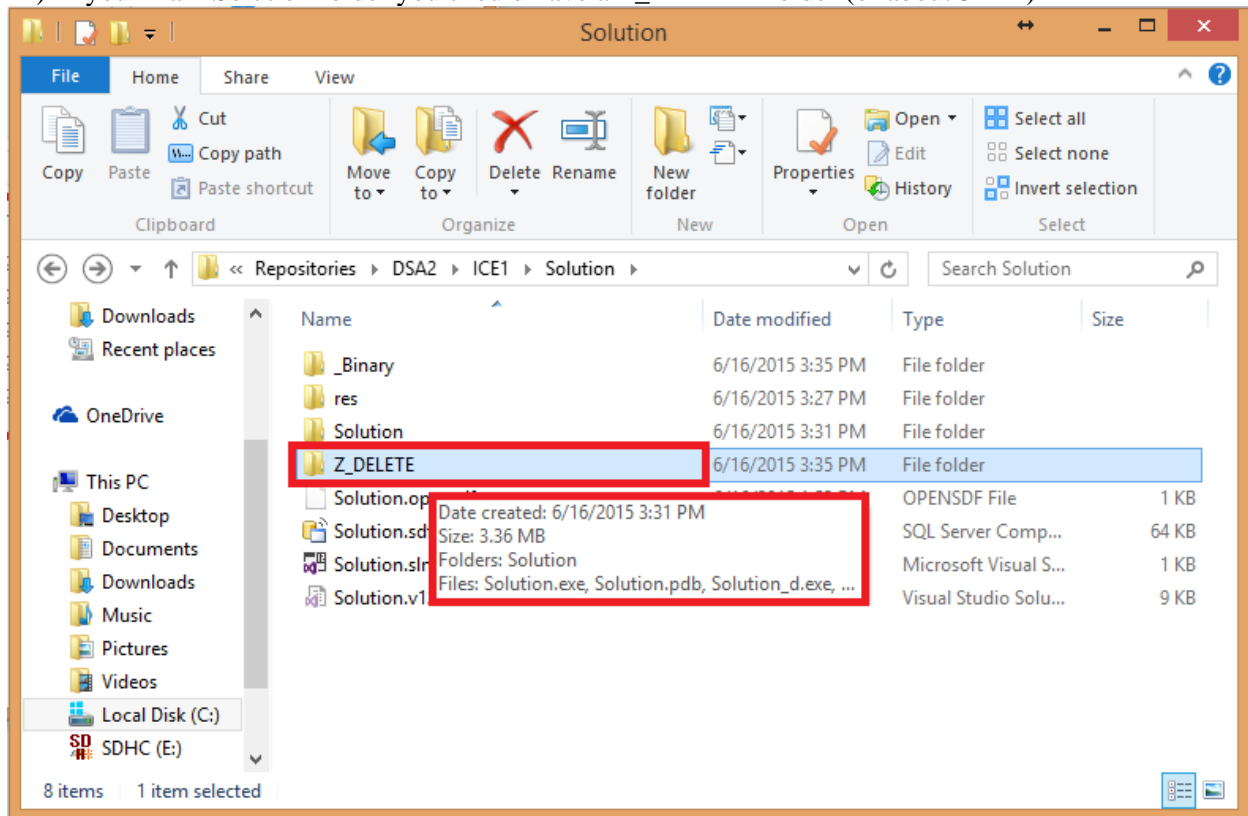The executable should be missing the _d now

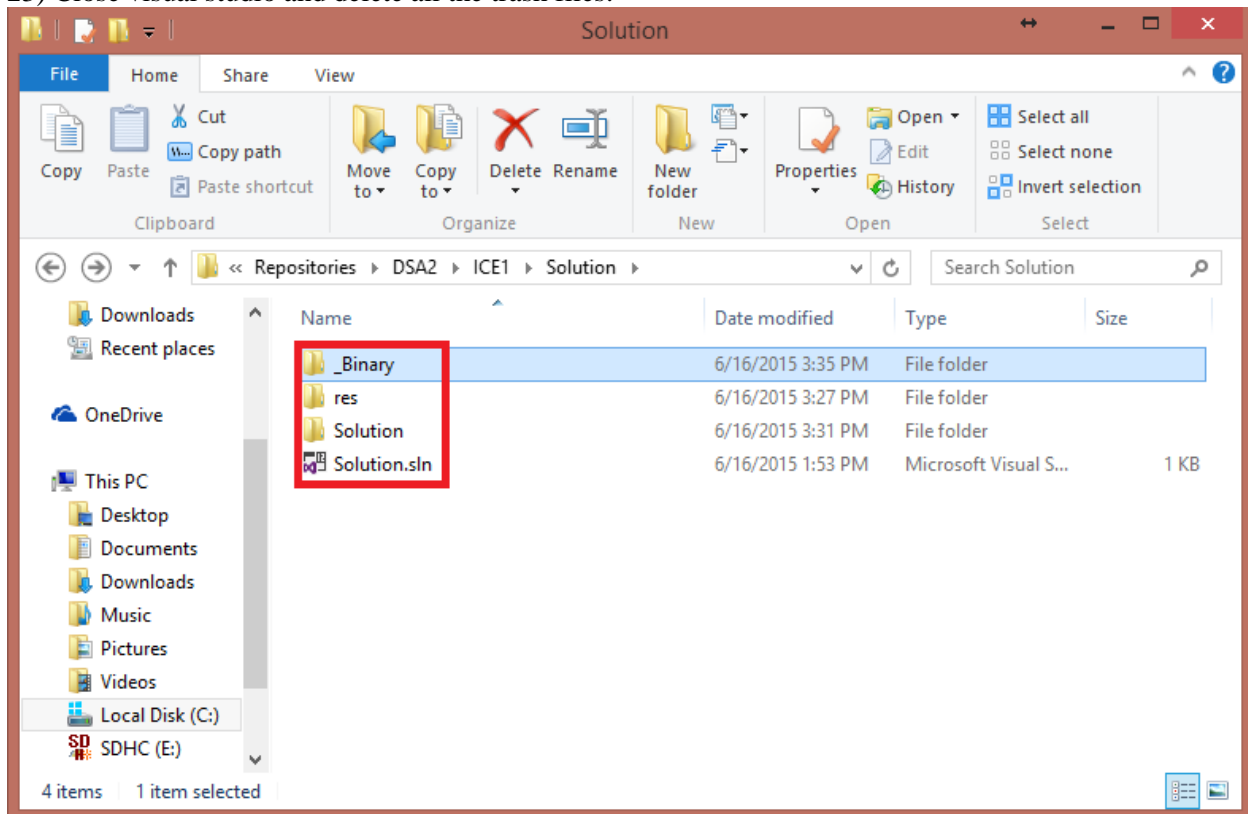20) in your _Binary folder you should only have two files:

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

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

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

24)   You should only have a Solution.sln file and those 3 folders (and a hidden .vs folder you may delete it or simply not zip it).

When you open an .sln file Visual studio is going to recreate the data base definition for your variables and functions (the .vs folder) this folder is used inside your configuration for your machine only, if you share this project the new computer will get rid of this folder 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 5 MB with 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.

25)   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)