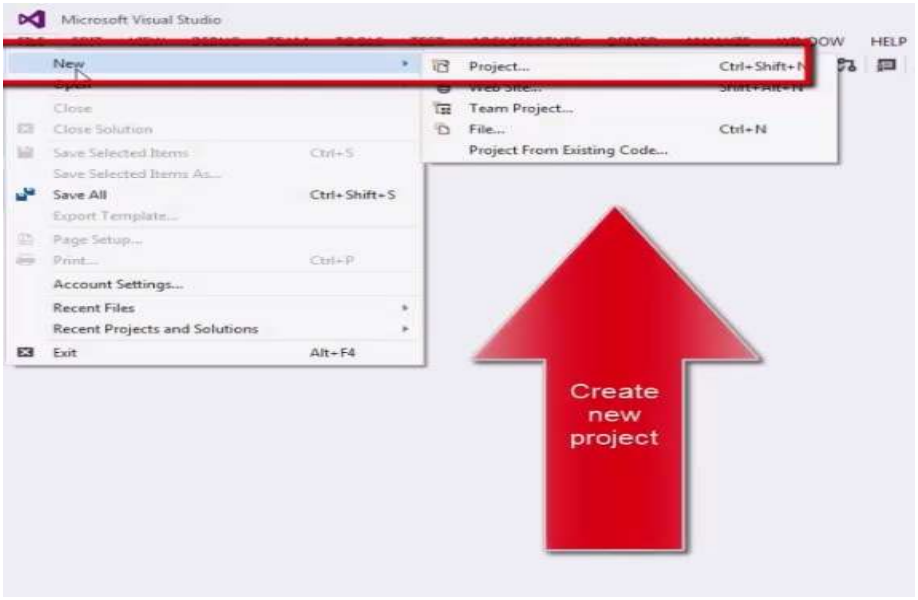
- DLL is a shared library that contains code and data that is used by more than one program at a time and like the executable file DLL cannot run directly.
- DLL (Dynamic-link library) is called by the application.

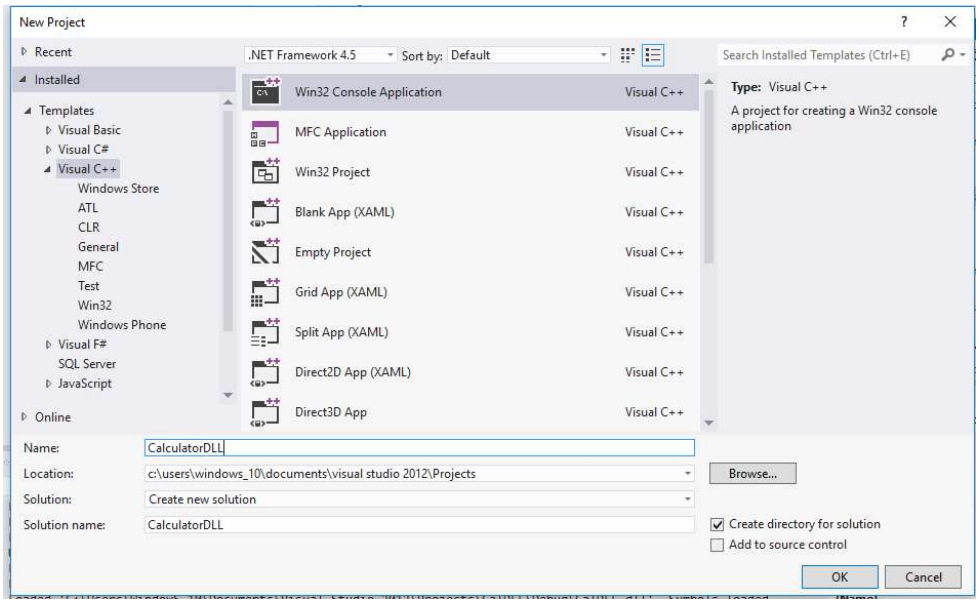
Steps to create DLL in C++

Here I will describe how to create a DLL project in C++ using the visual studio.

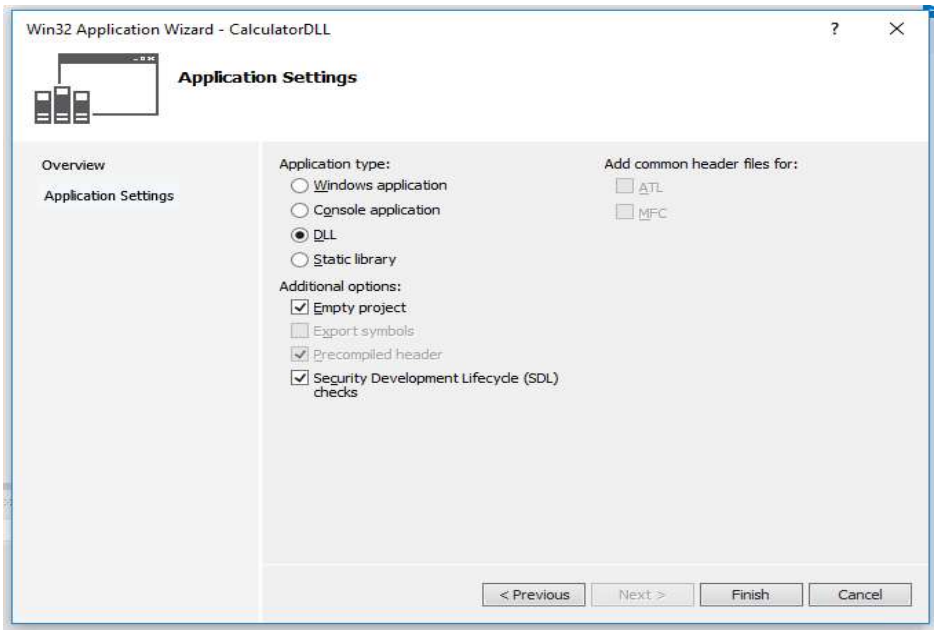
- Open the visual studio and click on the menu bar to create a new project. See the below Image.



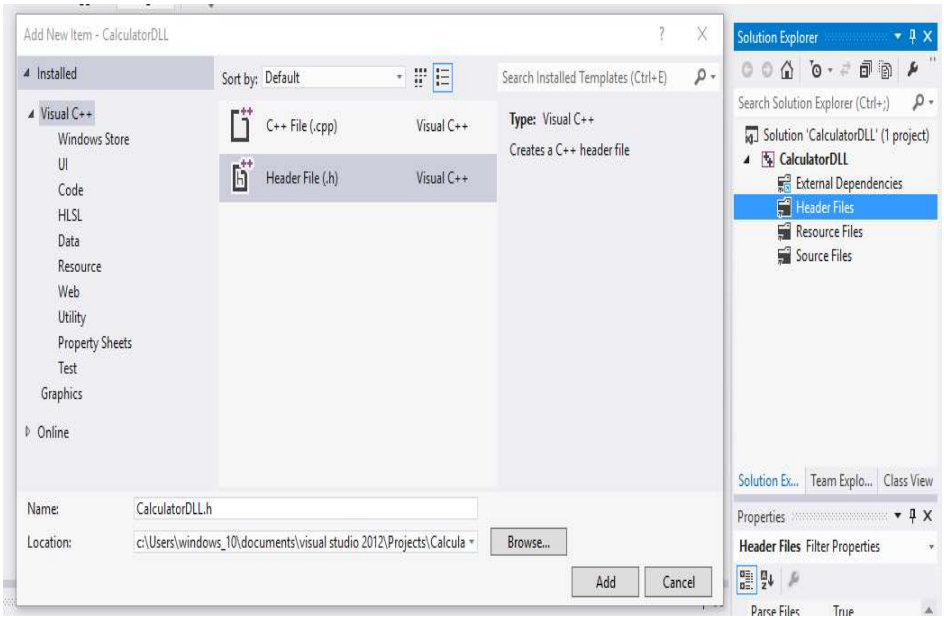
- After selecting the new project, a new dialog box will be open, here select the project type Win32 and give the name to the DLL project.



- On the Overview page of the Win32 Application Wizard dialog box, choose the Next button. After clicking the next button a new window will open. It is Application setting window here we will select the type of the application and click on the finish button to create the DLL project.



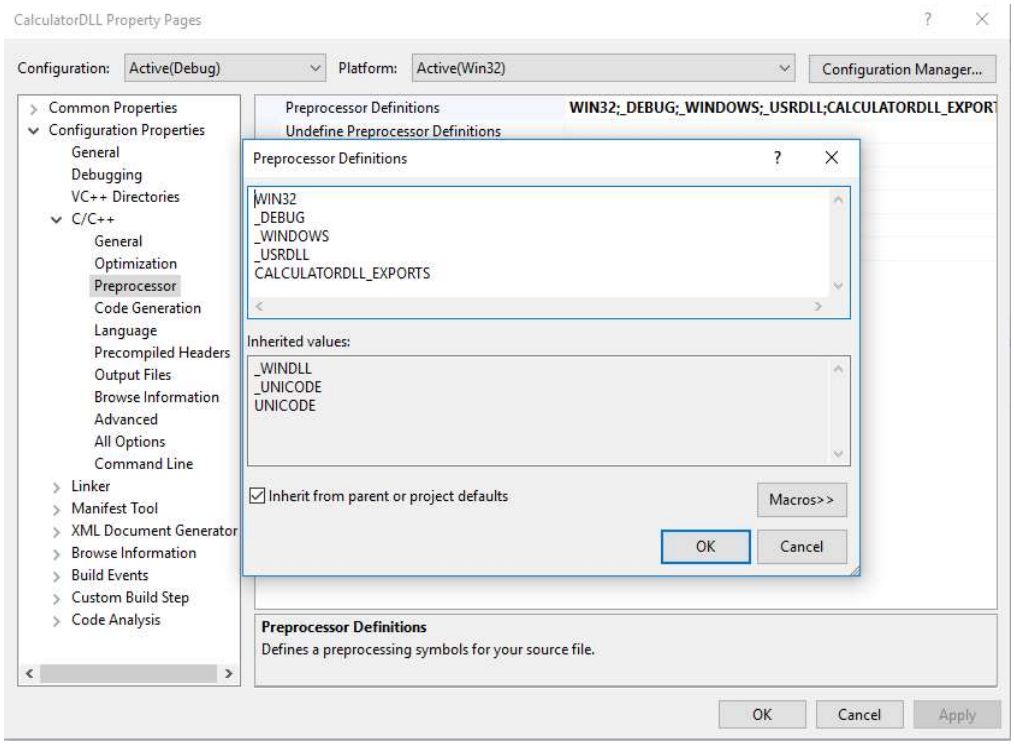
- After creating the DLL project you have to add the header files and source file as per your requirements. Here I am adding only one header file.



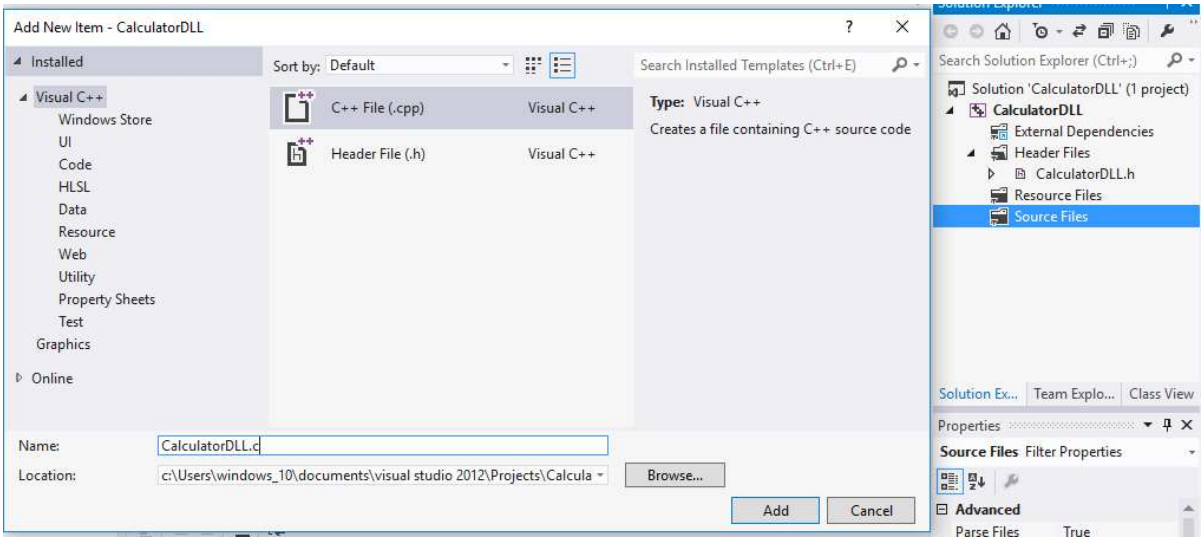
- When you have created the header file then write the desired content as per the requirements. Here I am creating a library that performs some basic arithmetic operation like addition.

```
#ifndef _CALCULATORDLL_h_
#define _CALCULATORDLL_h_
#ifdef CALCULATORDLL_EXPORTS
#define CALCULATORDLL_API __declspec(dllexport)
#else
#define CALCULATORDLL_API __declspec(dllimport)
#endif
CALCULATORDLL_API int Addition(int x,int y);
#endif
```

Note: When you have created a DLL project then automatically PROJECTNAME_EXPORTS is defined in preprocessor symbols of the DLL project. In this example, CALCULATIONDLL_EXPORTS is defined when your CALCULATIONDLL DLL project is built.



- Now it's time to define your class member function in the source file. Here I am defining all member functions in CalculatorDLL.C file.



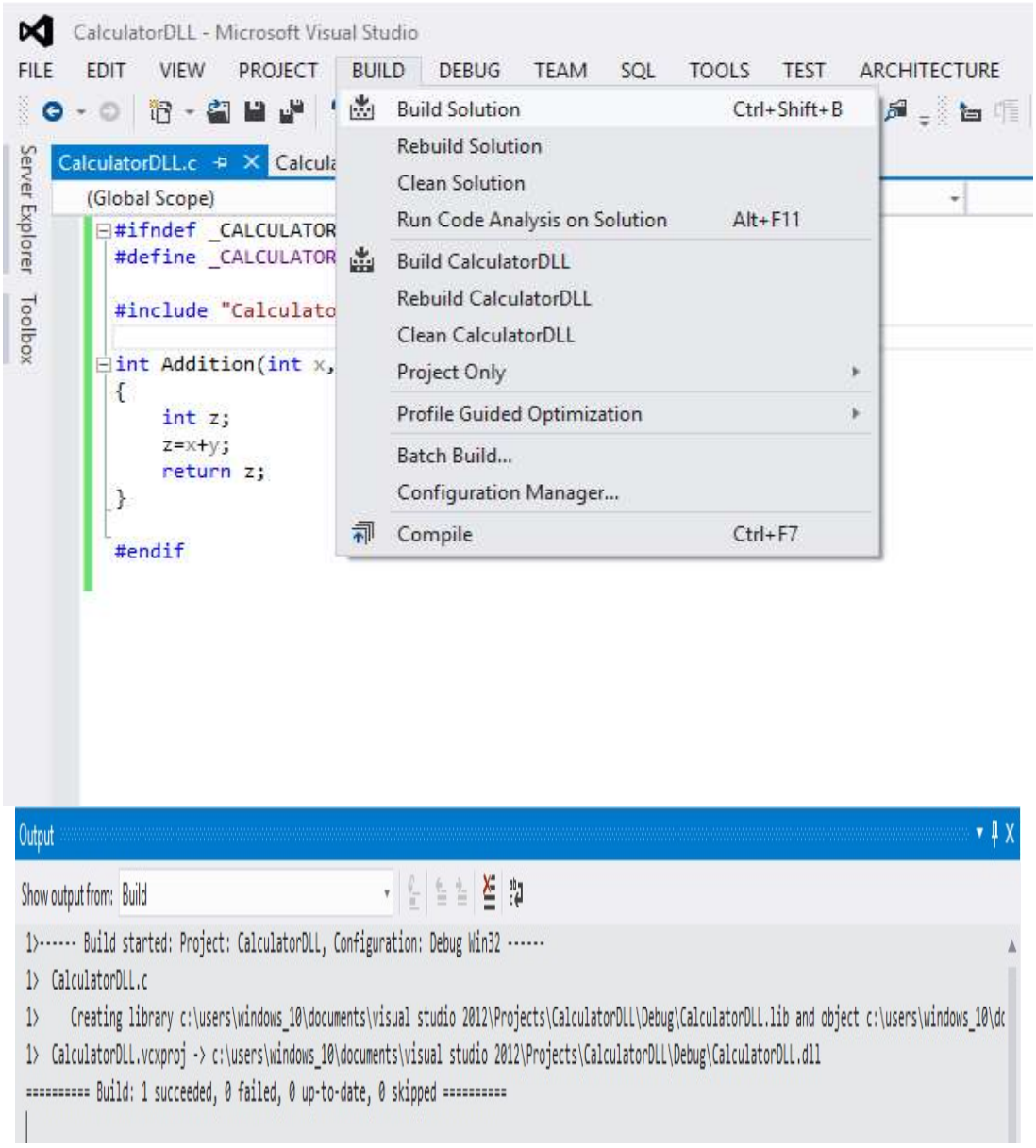

```
#ifndef _CALCULATORDLL_c_
#define _CALCULATORDLL_c_

#include "CalculatorDLL.h"

int Addition(int x,int y)
{
	int z;
	z=x+y;
	return z;
}


#endif
```

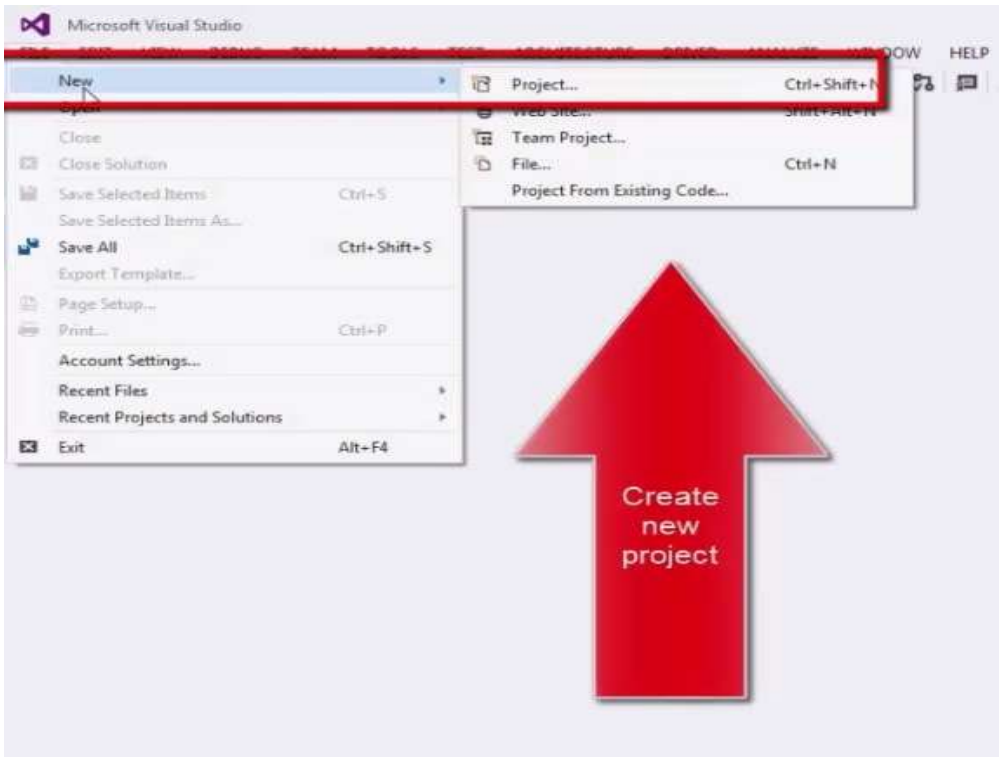
Now source and header files are added to the DLL project, to create the DLL and lib just build the DLL project. If everything is fine and your DLL project compiles perfectly without any error then a DLL and .lib file will be generated.



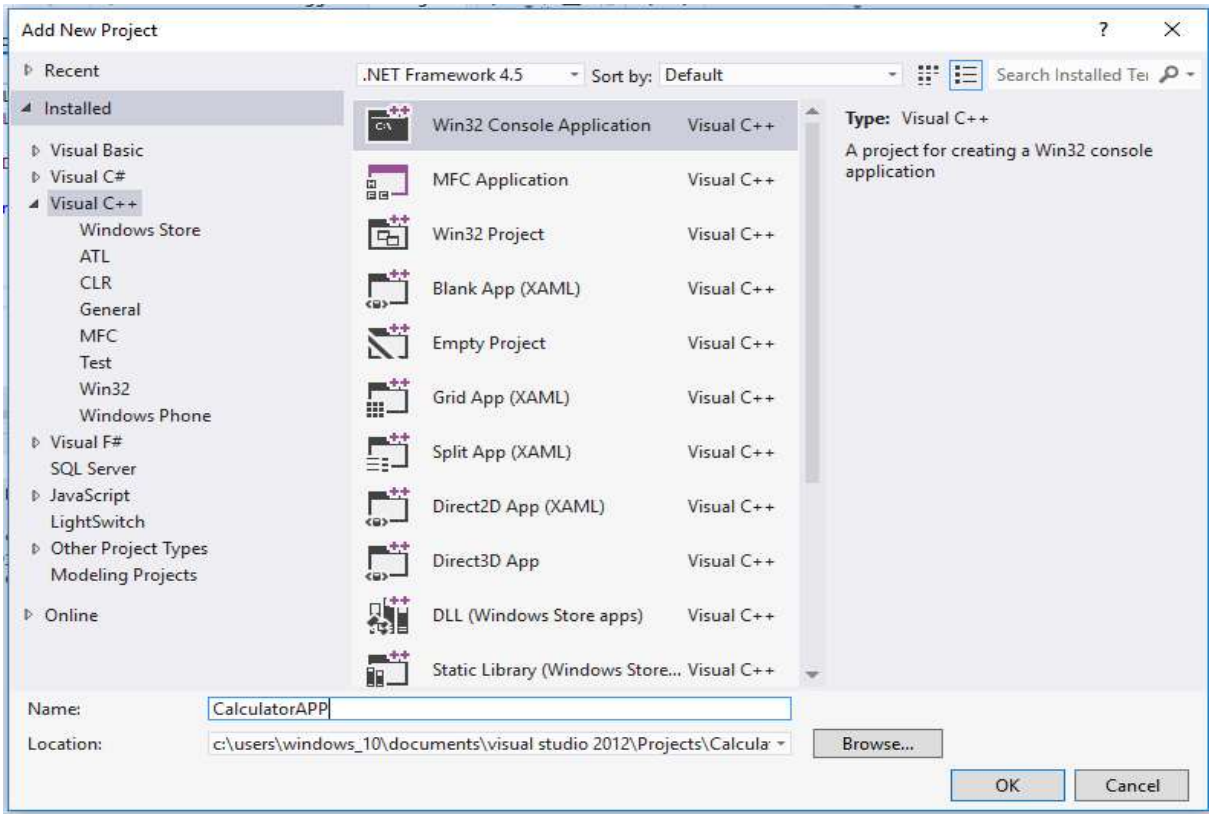
Steps to create a C ++ Application

Here I am creating a c++ application that will use the created DLL.

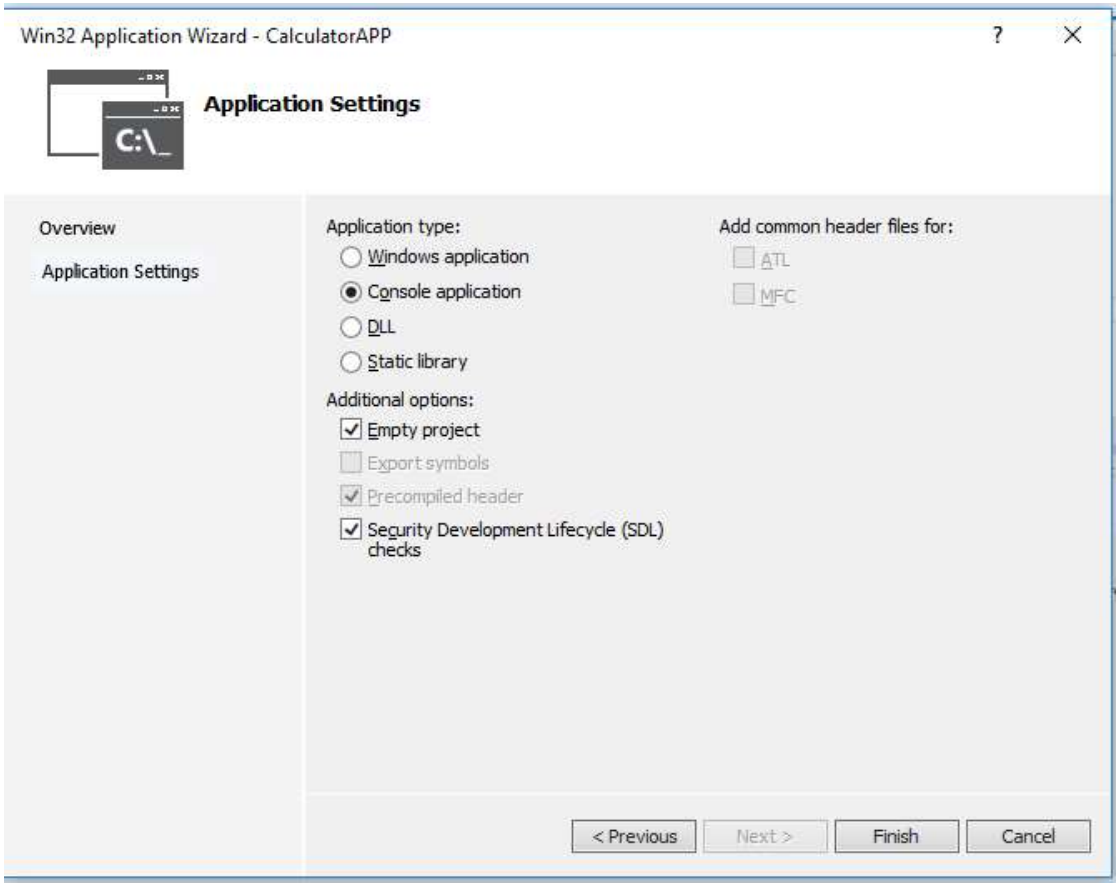
- Click on the menu bar to create a new c++ Application project that uses the DLL which I have created just now.



- After selecting the new project a new dialog box will be open, here select the project type Win32 Console Application and give the name to the App project.



- On the Overview page of the Win32 Application Wizard dialog box, choose the Next button. After clicking the next button a new window will open. It is the Application setting window here we will select the type of the application and click on the finish button to create the c++ Console Application project.

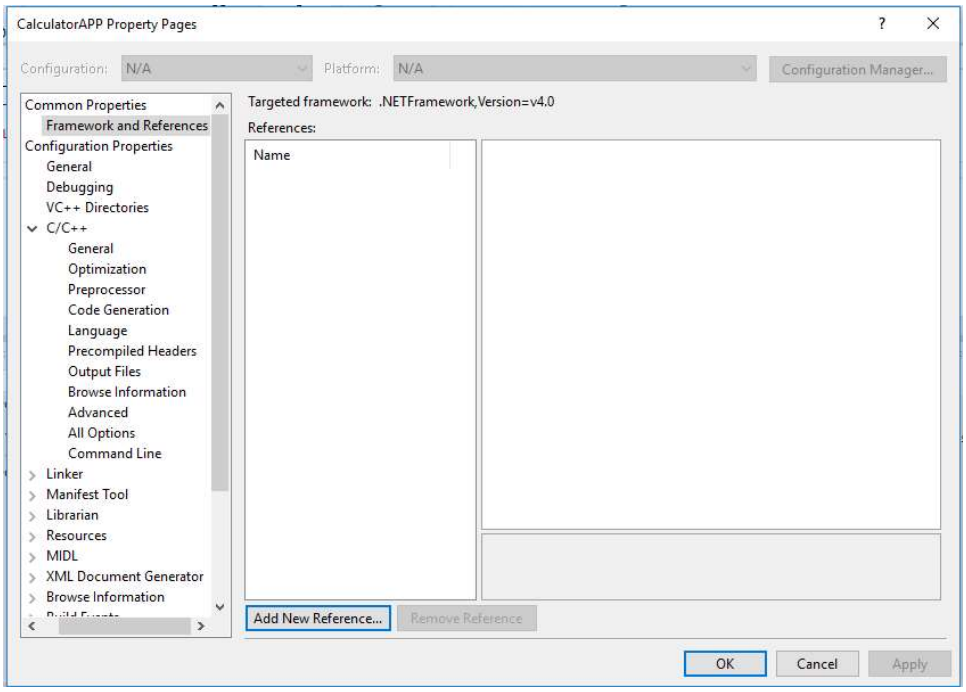


Now your C++ application project is ready to use the DLL (Dynamic linking library).

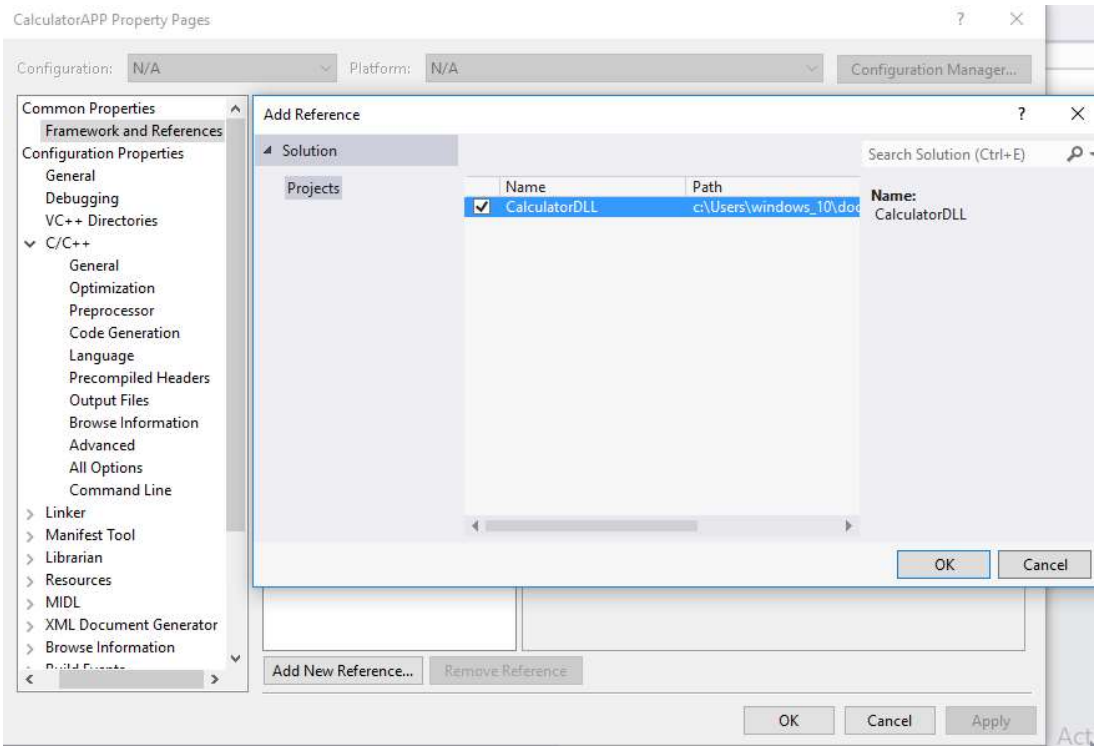
How to Link DLL with c++ Application

Here I am discussing simple steps to link the DLL project with the C++ Application project.

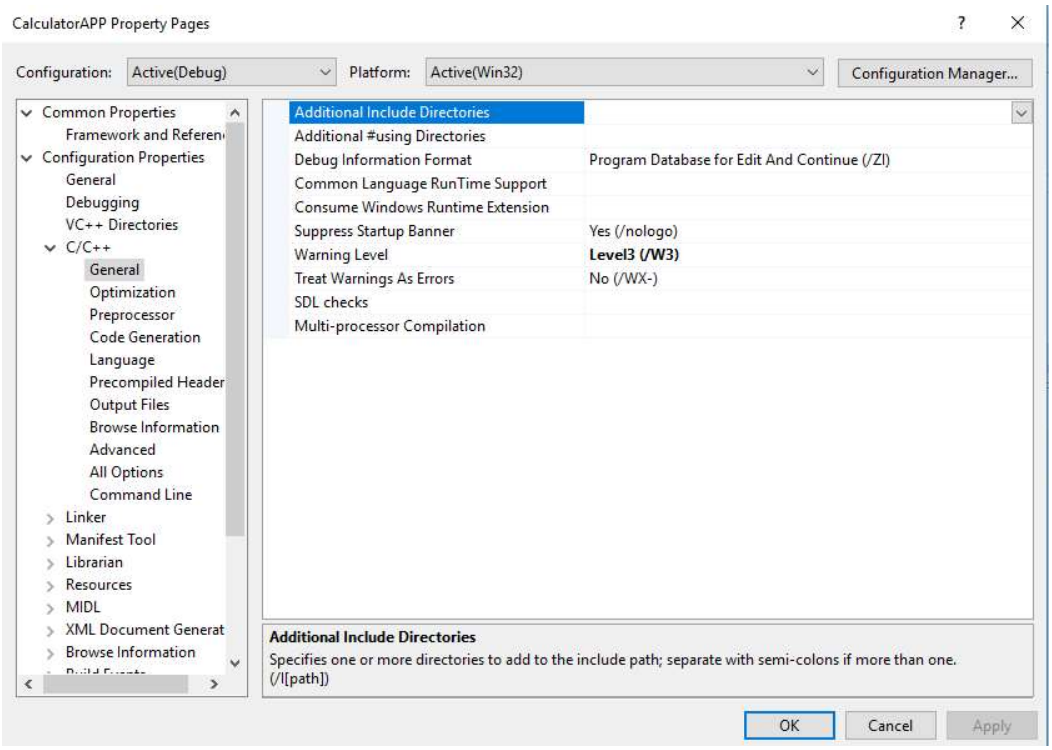
- When we have created the DLL and Application then after that we have to reference the DLL to the Application that makes the enable to Application to use the DLL function as per the requirement. To do this, under the CalculatorAPP project in Solution Explorer, select the References item. On the menu bar, choose Project, Add Reference.

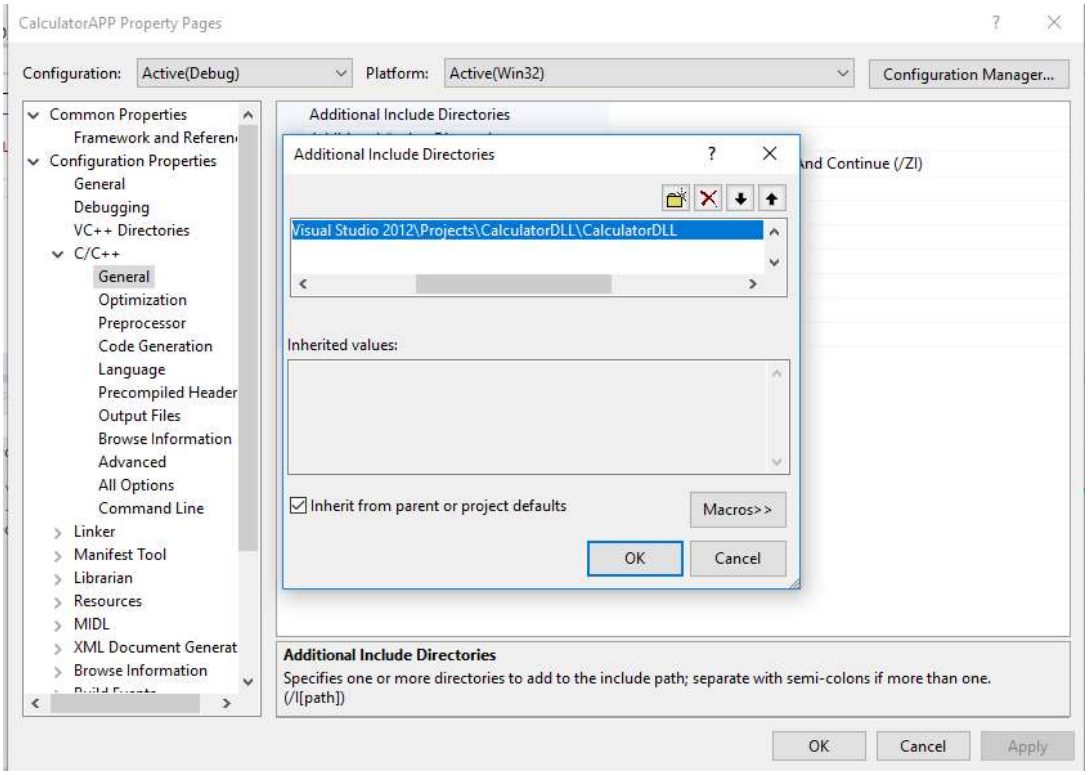


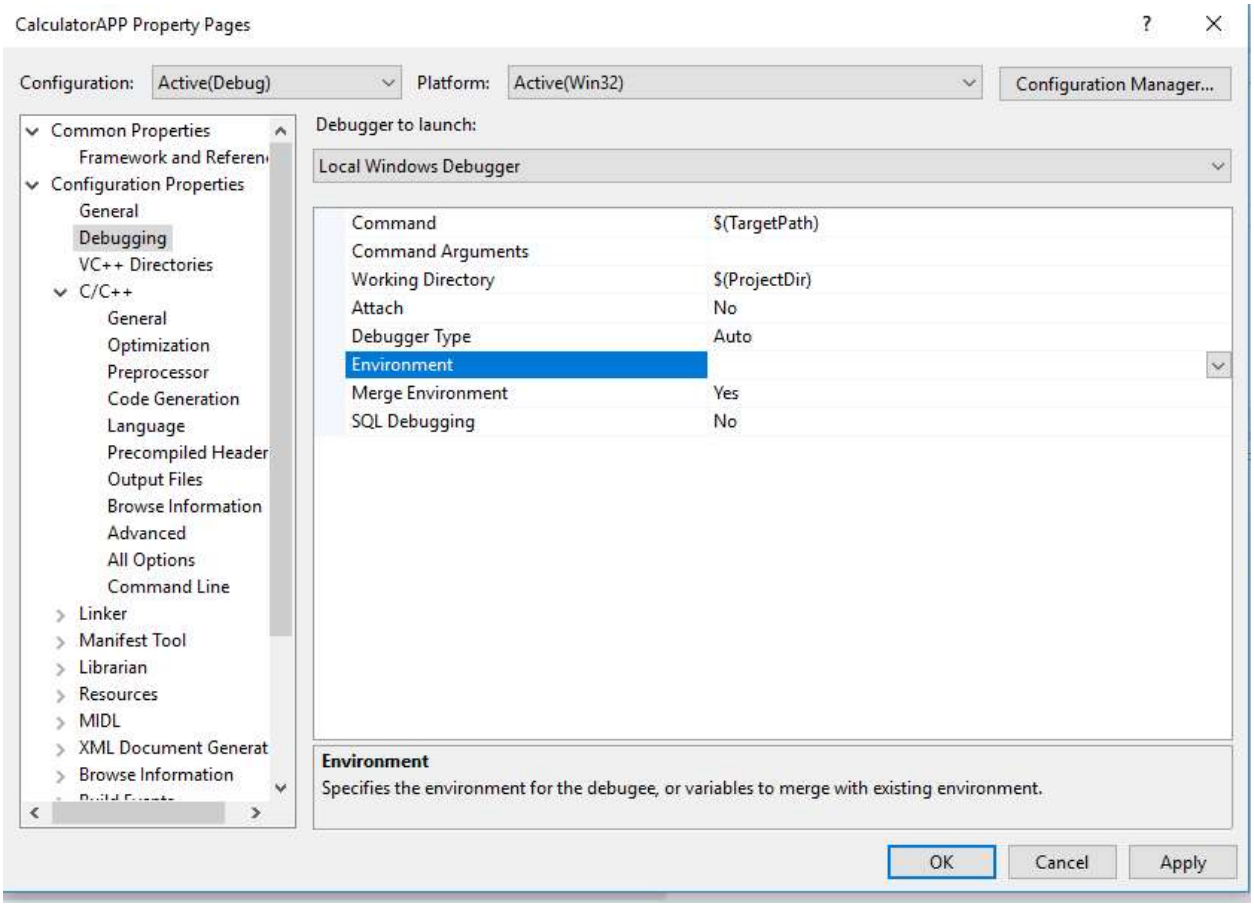
- When you click on the Add new Reference then a dialog box will be open which has the lists of the library that you can reference. You need to just click on the check button to the required library. Here only one library is showing in the dialog box.

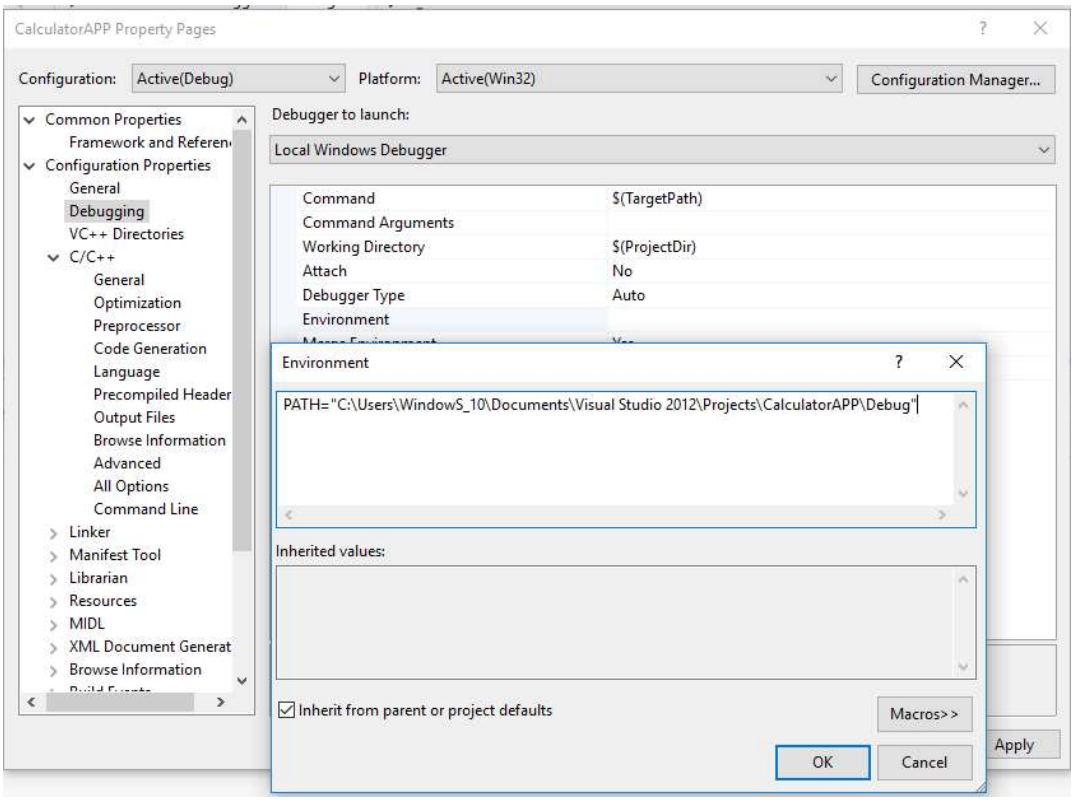


- Now your created library is linked with the created Application, but before using the DLL in Application you have to add the DLL header file. We just reference the DLL header file to give the path of original DLL header files in Application project included directories path.

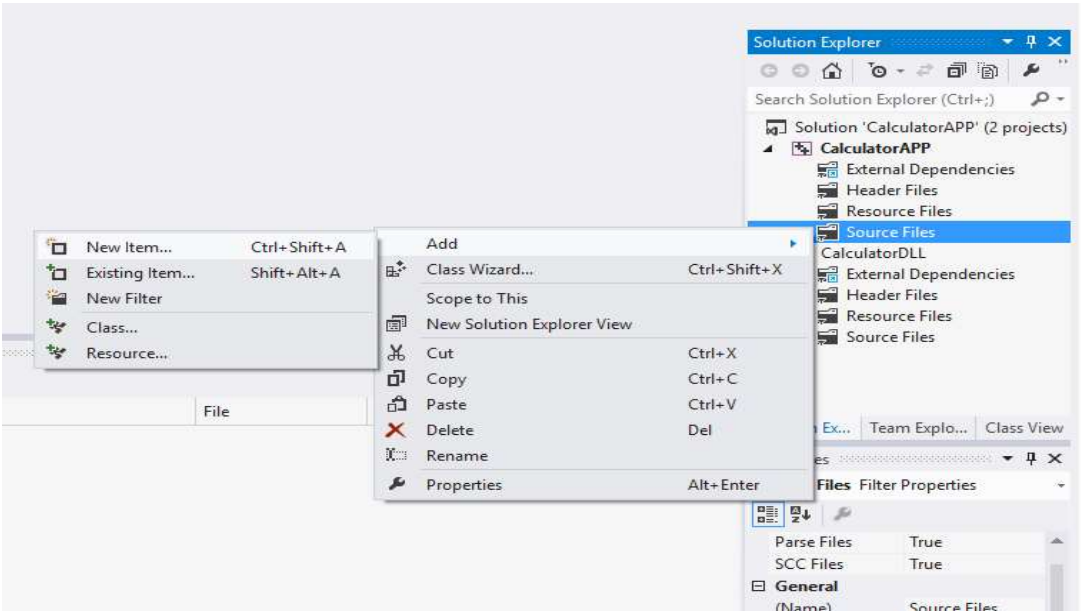


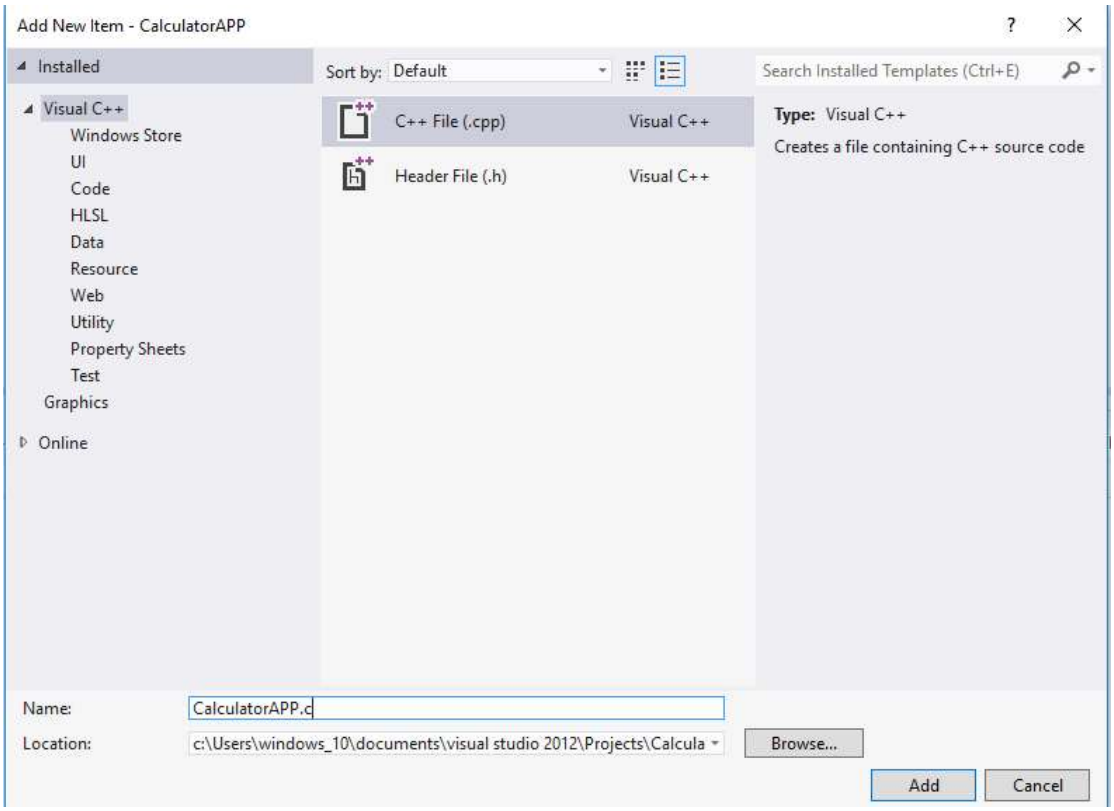



CalculatorAPP Property Pages
Configuration: Active(Debug)
Platform: Active(Win32)
Configuration Manager...
Common Properties
Framework and Referen
Configuration Properties
General
Debugging
VC++ Directories
C/C++
General
Optimization
Preprocessor
Code Generation
Language
Precompiled Header
Output Files
Browse Information
Advanced
All Options
Command Line
Linker
Manifest Tool
Librarian
Resources
MIDL
XML Document Generat
Browse Information
Debugger to launch:
Local Windows Debugger
Command
$(TargetPath)
Command Arguments
Working Directory
$(ProjectDir)
Attach
No
Debugger Type
Auto
Environment
Merge Environment
Yes
SQL Debugging
No
Environment
Specifies the environment for the debugee, or variables to merge with existing environment.
OK
Cancel
Apply



- Now it's time to define your class member function in the source file. Here I am calling all member functions in CalculatorAPP.C file.

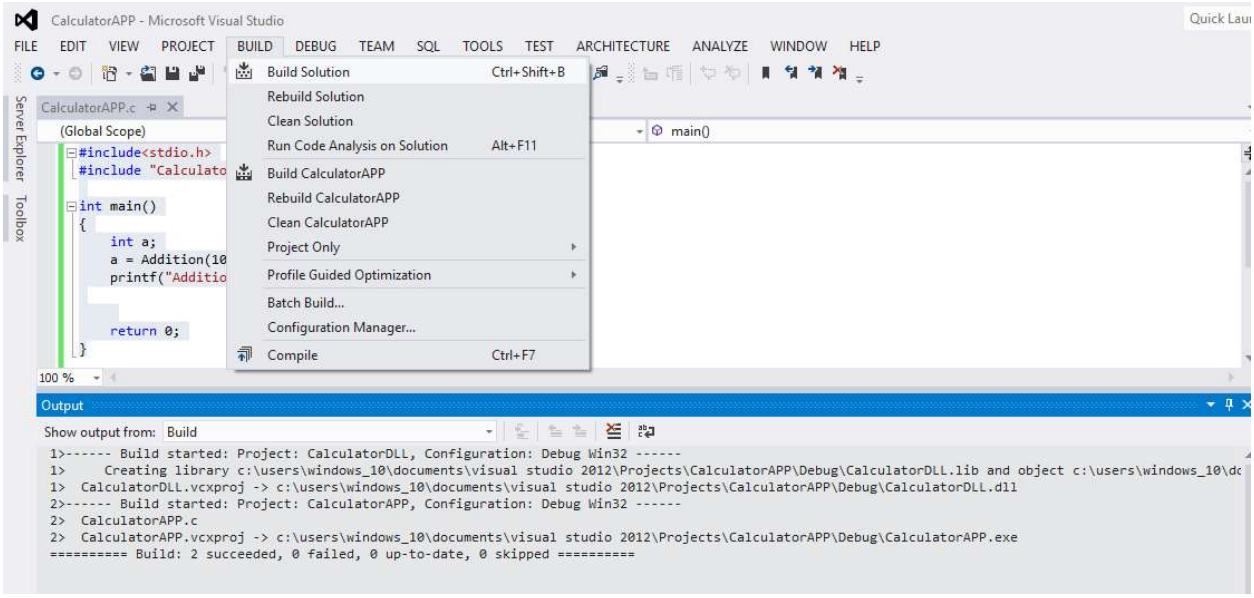


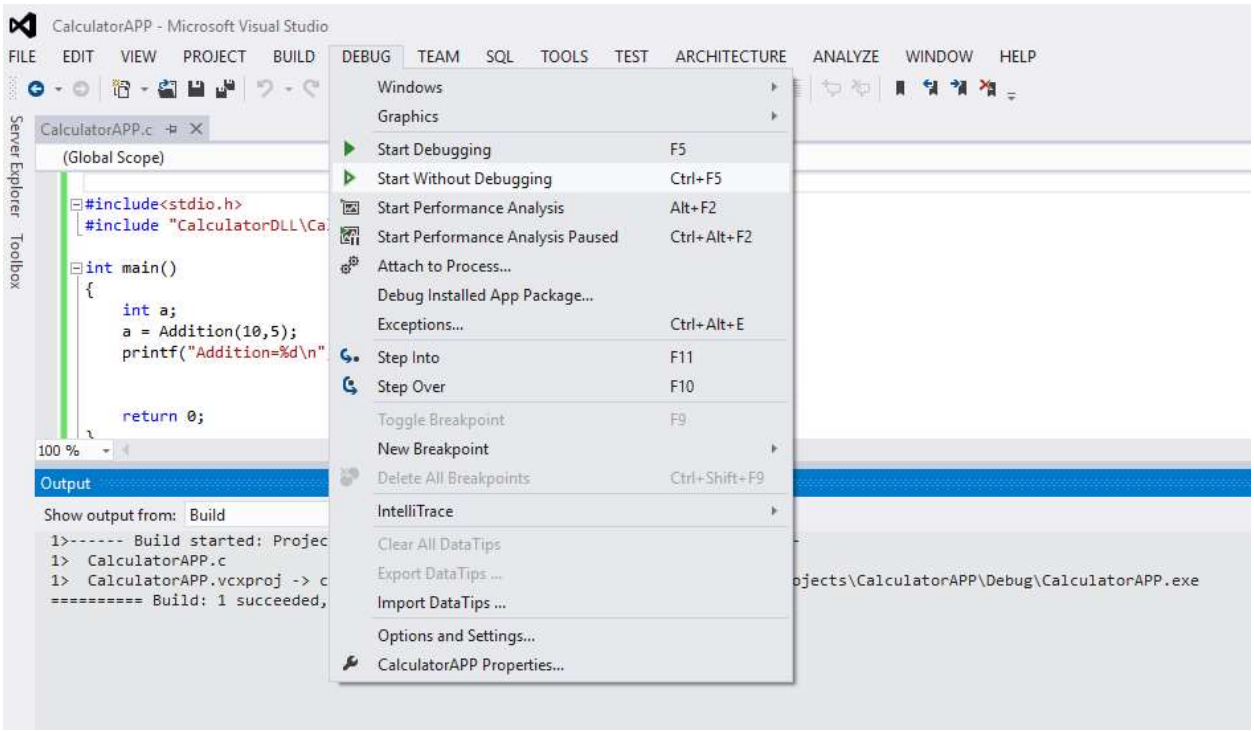


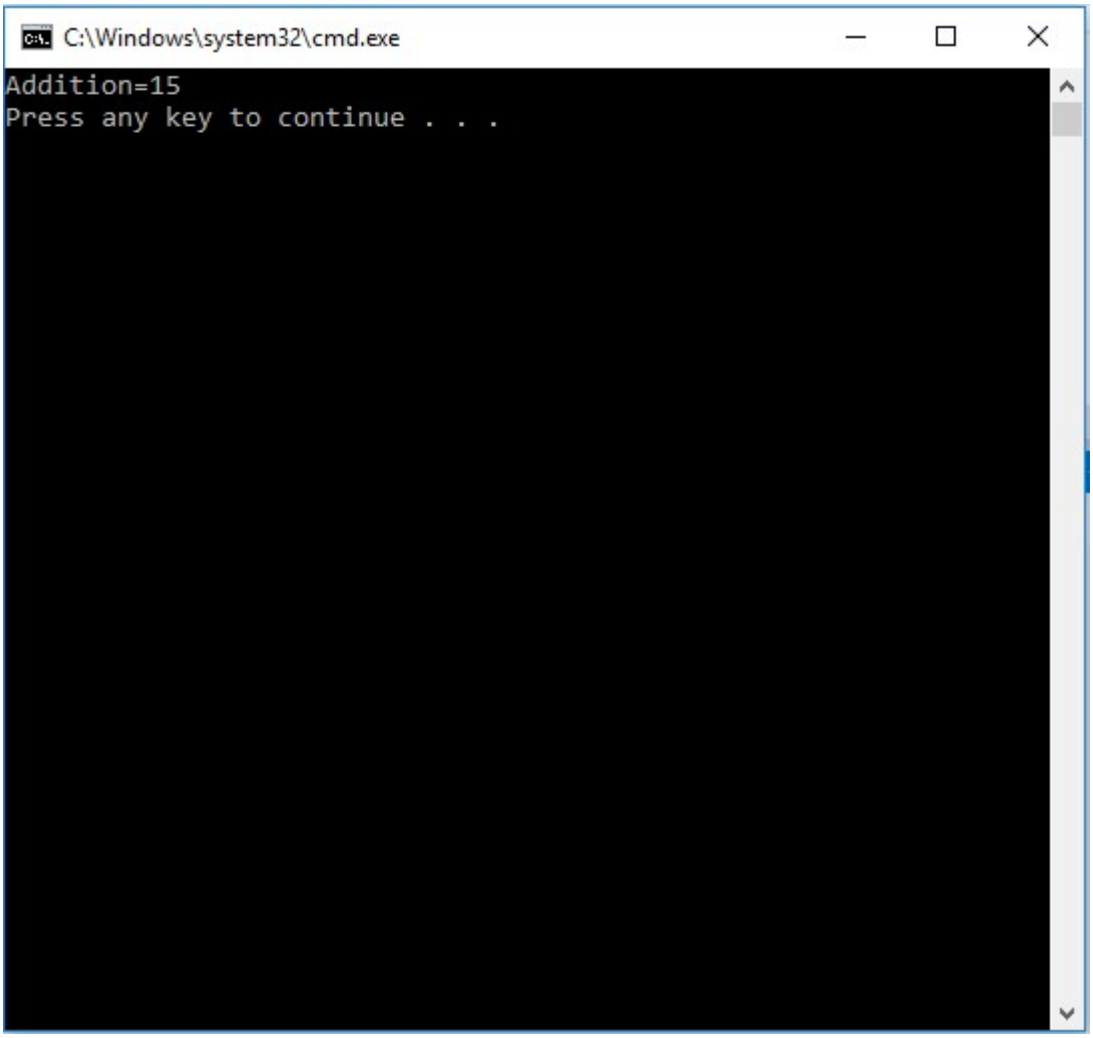

```
#include<stdio.h>
#include "CalculatorDLL\CalculatorDLL.h"

int main()
{
	int a;
	a = Addition(10,5);
	printf("Addition=%d\n",a);


	return 0;
}
```


CalculatorAPP - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM SQL TOOLS TEST ARCHITECTURE ANALYZE WINDOW HELP
Build Solution Ctrl+Shift+B
Rebuild Solution
Clean Solution
Run Code Analysis on Solution Alt+F11
Build CalculatorAPP
Rebuild CalculatorAPP
Clean CalculatorAPP
Project Only
Profile Guided Optimization
Batch Build...
Configuration Manager...
Compile Ctrl+F7
CalculatorAPP.c
(Global Scope)
main()
#include<stdio.h>
#include "Calculato
int main()
{
int a;
a = Addition(10
printf("Additio
return 0;
}
100 %
Output
Show output from: Build
1>------ Build started: Project: CalculatorDLL, Configuration: Debug Win32 ------
1> Creating library c:\users\windows_10\documents\visual studio 2012\Projects\CalculatorAPP\Debug\CalculatorDLL.lib and object c:\users\windows_10\dc
1> CalculatorDLL.vcxproj -> c:\users\windows_10\documents\visual studio 2012\Projects\CalculatorAPP\Debug\CalculatorDLL.dll
2>------ Build started: Project: CalculatorAPP, Configuration: Debug Win32 ------
2> CalculatorAPP.c
2> CalculatorAPP.vcxproj -> c:\users\windows_10\documents\visual studio 2012\Projects\CalculatorAPP\Debug\CalculatorAPP.exe
========== Build: 2 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========


CalculatorAPP - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM SQL TOOLS TEST ARCHITECTURE ANALYZE WINDOW HELP
Windows
Graphics
Start Debugging F5
Start Without Debugging Ctrl+F5
Start Performance Analysis Alt+F2
Start Performance Analysis Paused Ctrl+Alt+F2
Attach to Process...
Debug Installed App Package...
Exceptions... Ctrl+Alt+E
Step Into F11
Step Over F10
Toggle Breakpoint F9
New Breakpoint
Delete All Breakpoints Ctrl+Shift+F9
IntelliTrace
Clear All DataTips
Export DataTips ...
Import DataTips ...
Options and Settings...
CalculatorAPP Properties...
CalculatorAPP.c
(Global Scope)
#include<stdio.h>
#include "CalculatorDLL\Ca
int main()
{
int a;
a = Addition(10,5);
printf("Addition=%d\n"
return 0;
}
100 %
Output
Show output from: Build
1>------ Build started: Projec
1> CalculatorAPP.c
1> CalculatorAPP.vcxproj -> c
ojects\CalculatorAPP\Debug\CalculatorAPP.exe
========== Build: 1 succeeded,


C:\Windows\system32\cmd.exe
Addition=15
Press any key to continue . . .