Visual Studio Guide

M Gouri Sankar

Installation

- Download the VisualStudioSetup.exe (Community Edition)
- Run the Installer
- Select the necessary Visual Studio Workloads for C++ development
  - **Desktop Development with C++**

    Used to build modern C++ apps for Windows using various tools
  - **Universal Windows Platform Development**

    Used to create application for the Universal Windows Platform with C#, C++, VB etc.
- Also, select the optional individual components, necessary for your development.
- Modify the installation path, if necessary
- Install, the selected components and workloads.
- The installer also helps to modify Visual Studio by adding and removing components as necessary.
- Launch Visual Studio

New Project

- Select a suitable template for your project

- Ensure that the language and platforms are chosen correctly, use the filters if necessary.

- Choose a name and location for your project
- The templates of your recent projects are always pinned for easier access.

Standard Directory Layout for Project

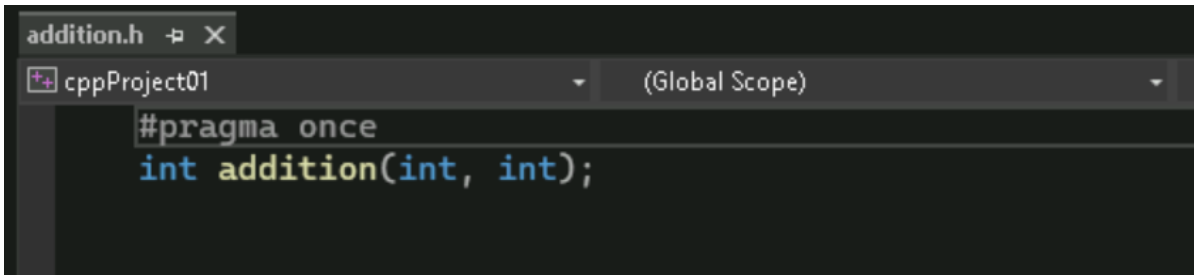
- **bin -** To place the executable file
- **src -** To place the source cpp files
- **obj -** To place the intermediate object file
- **inc/include -** To place the header files

- **lib:** To place the library files

Writing a simple C++ application

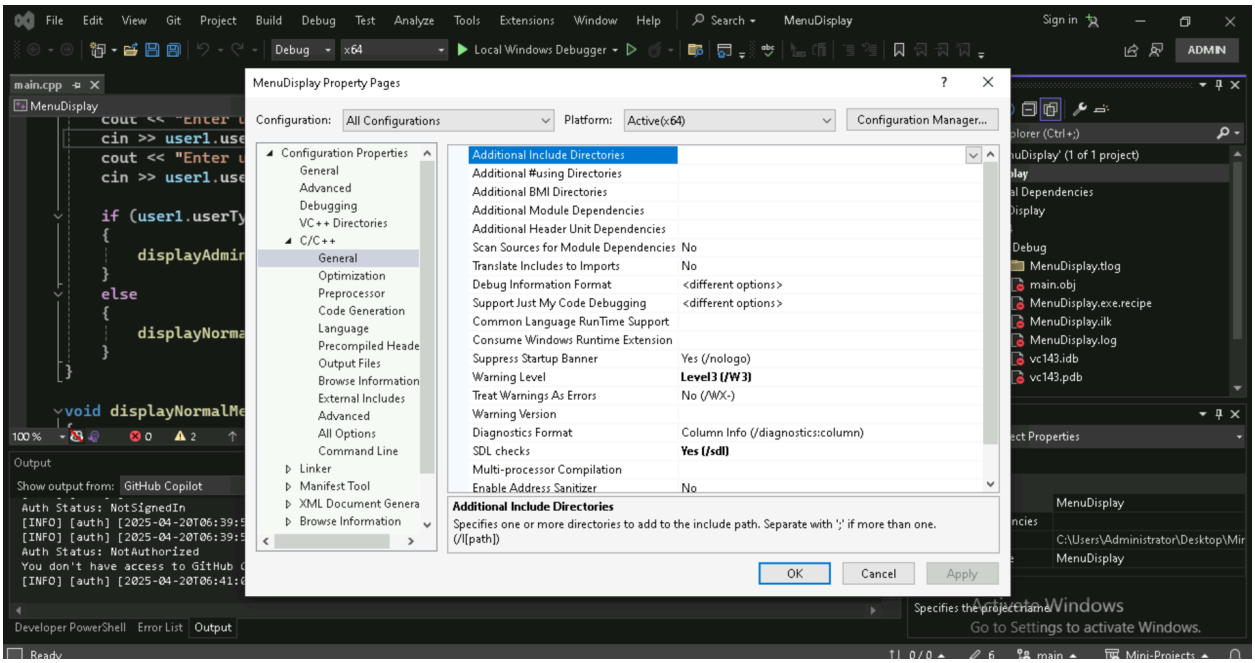
Function Declaration

- Identify the major functions required for the project/application
- In the **inc directory**, generate **.h header files** containing the declarations of the necessary functions.
- Remember to include **'# pragma once'** to ensure that the functions are declared only once even when the header files are used multiple times.



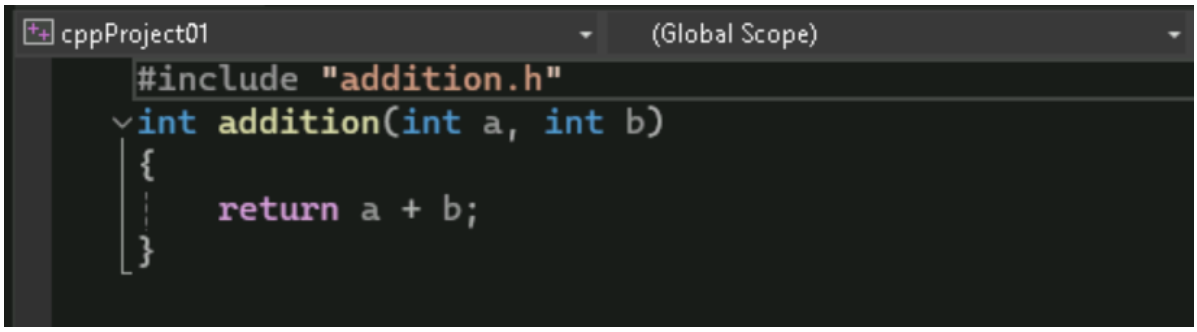
Setting the Additional Include files (.h location)

- Navigate to **Project Settings > C/C++ > General > Additional Include Directories**
- Use Macros to set where your custom header files are located for linking (inc folder)



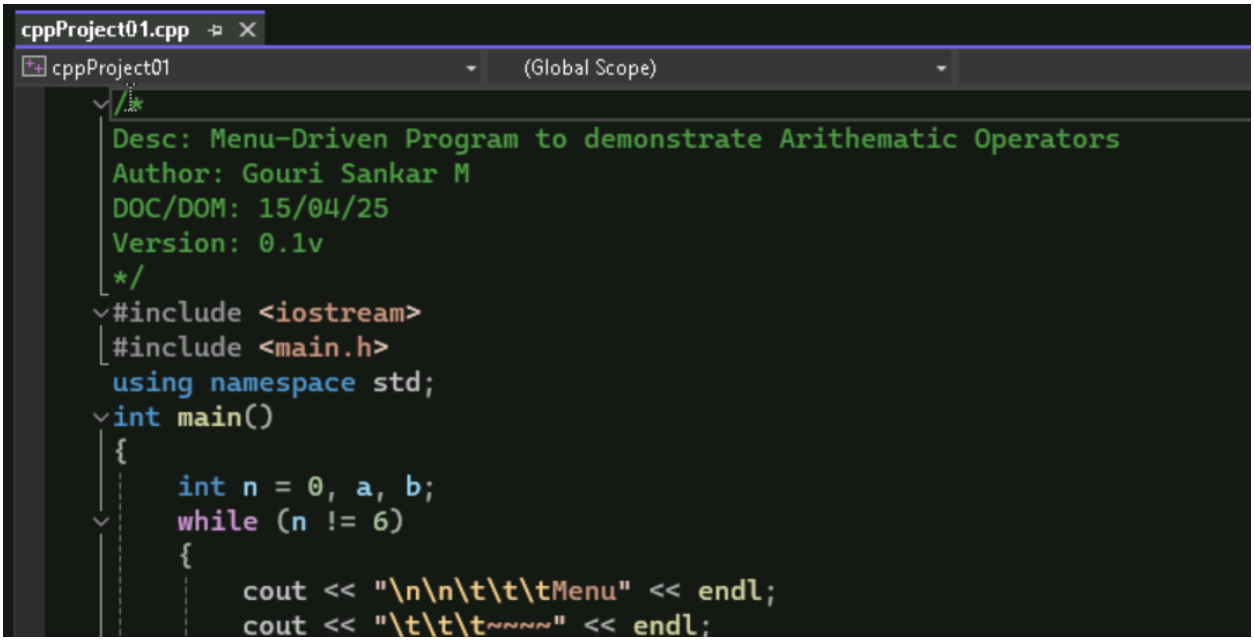
Function Definition

- Write the appropriate definition for the declared functions, in **.cpp** files.
- These function source code **.cpp** files are to be included in the **src** directory.
- Remember to include the header files containing the function declaration.



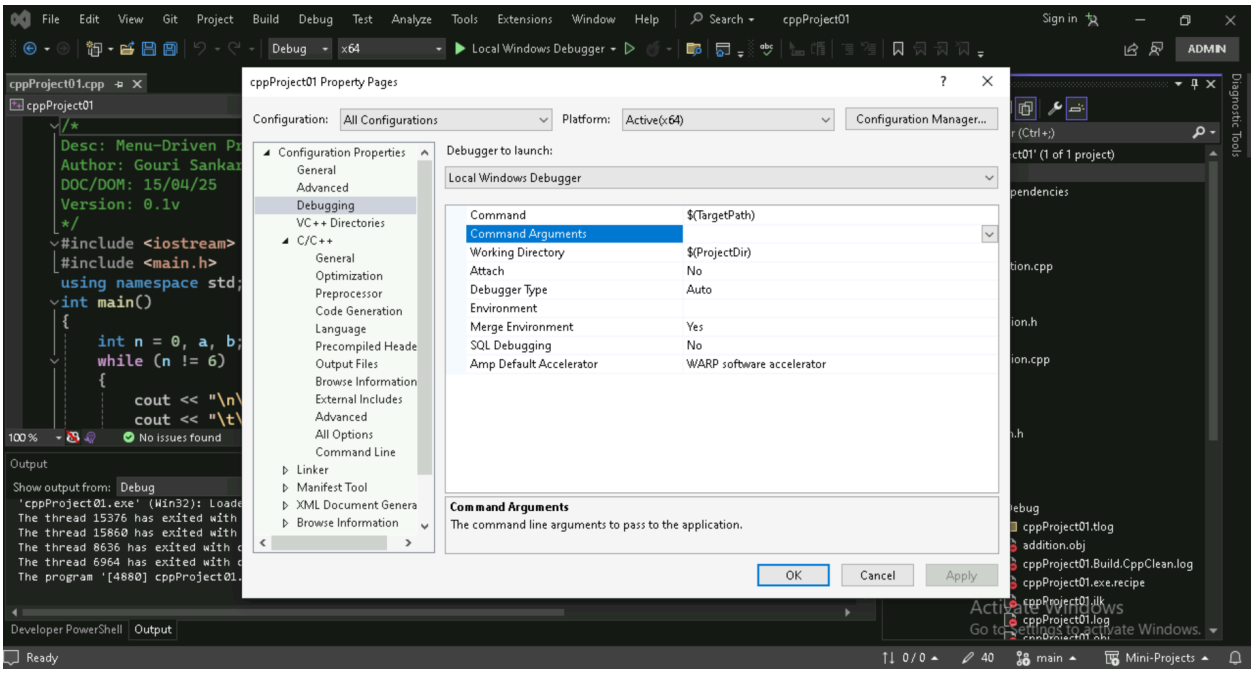
Driver Function (main function)

- Write the code for the main function that drives the application.
- The **.cpp** file is to be included in the appropriate **.src** directory
- Include all the header files, both standard and custom ones.
- Add any other necessary preprocessor directives.



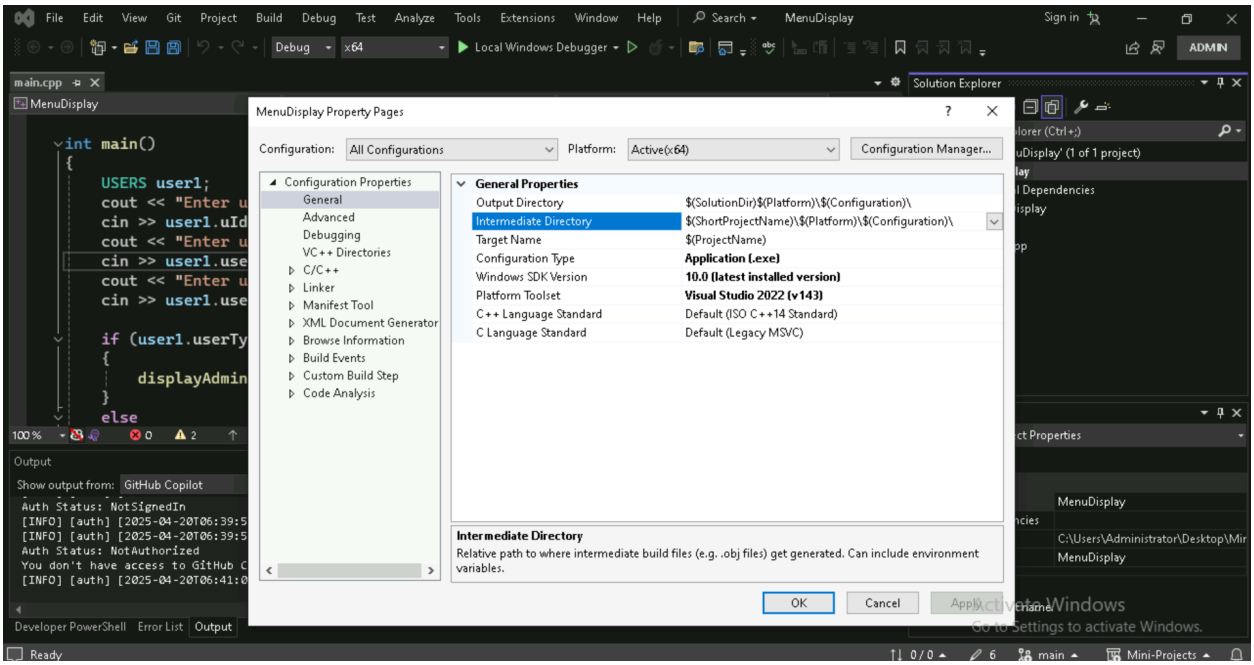
Command Line Arguments

- If your main function accepts command line arguments, you need to configure Visual Studio to provide those arguments.
- Navigate to **Project Settings > Configuration Properties > Debugging > Command Arguments**



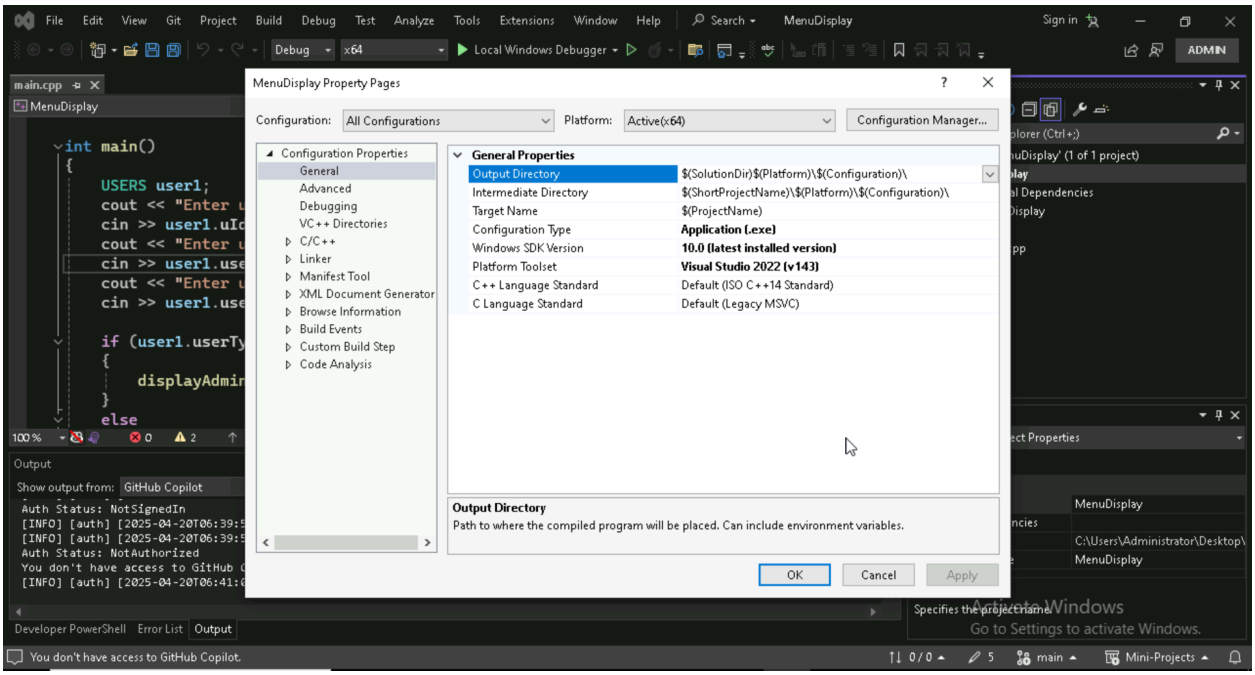
Setting the Intermediate Directory (.obj location)

- Navigate to **Project Settings > Configuration Properties > General > Intermediate Directory**
- Use Macros to set where your intermediate object files are to be generated (obj folder)



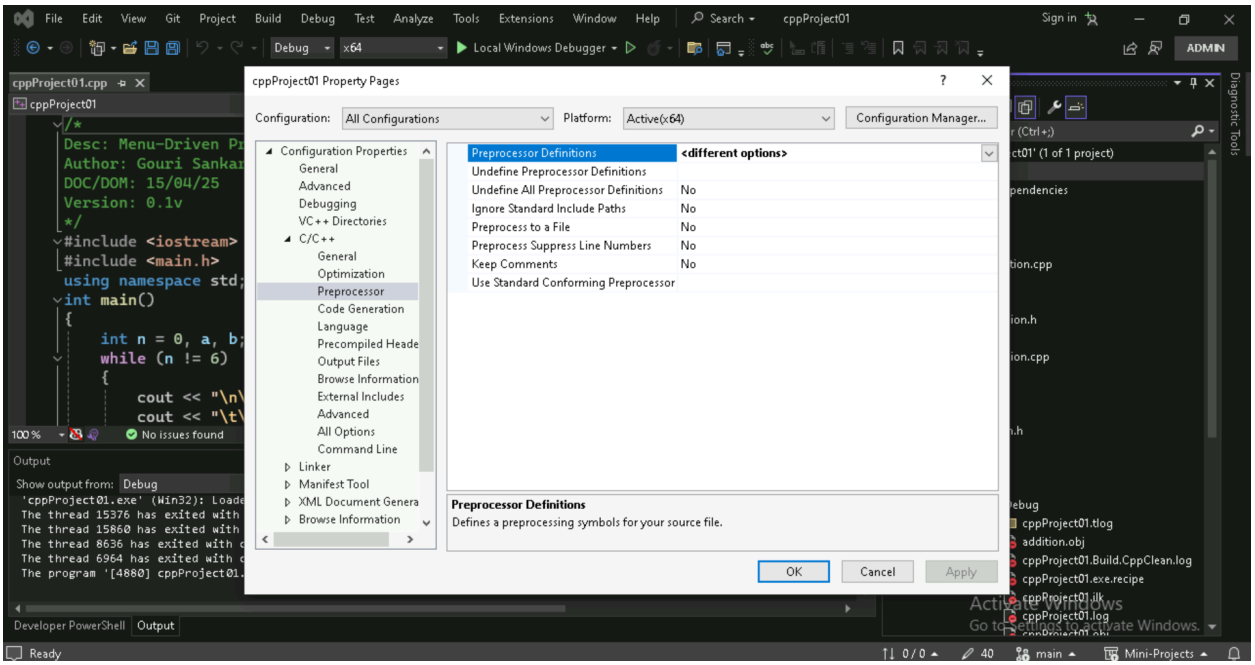
Setting the Output Directory (.exe location)

- Navigate to **Project Settings > Configuration Properties > General > Output Directory**
- Use Macros to set where your output executable is to be generated (bin folder)



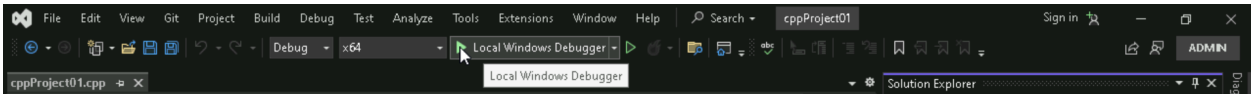
Preprocessor Definitions

- Navigate to **Project Settings > C/C++ > Preprocessor > Preprocessor Definitions**
- Add any extra Preprocessor definitions, if included in the application.



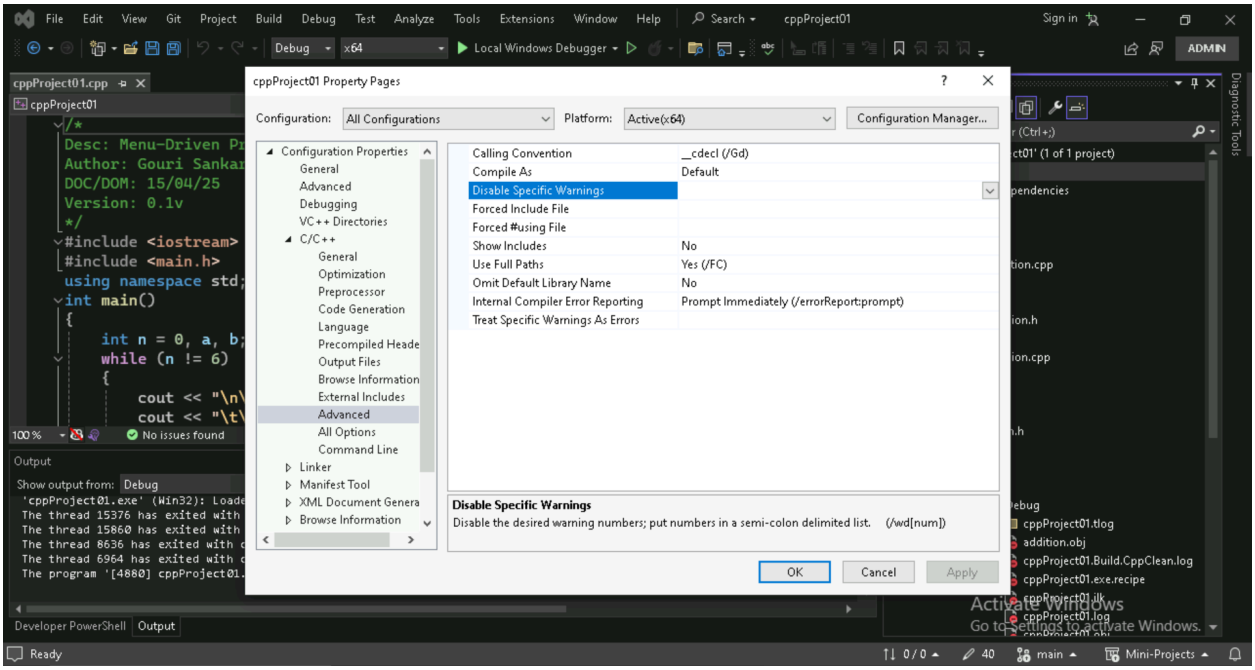
Debug the Program

- Select the **Configuration (Debug/Release)** and the **Platform**.
- Debug the program with **Local Windows Debugger.**



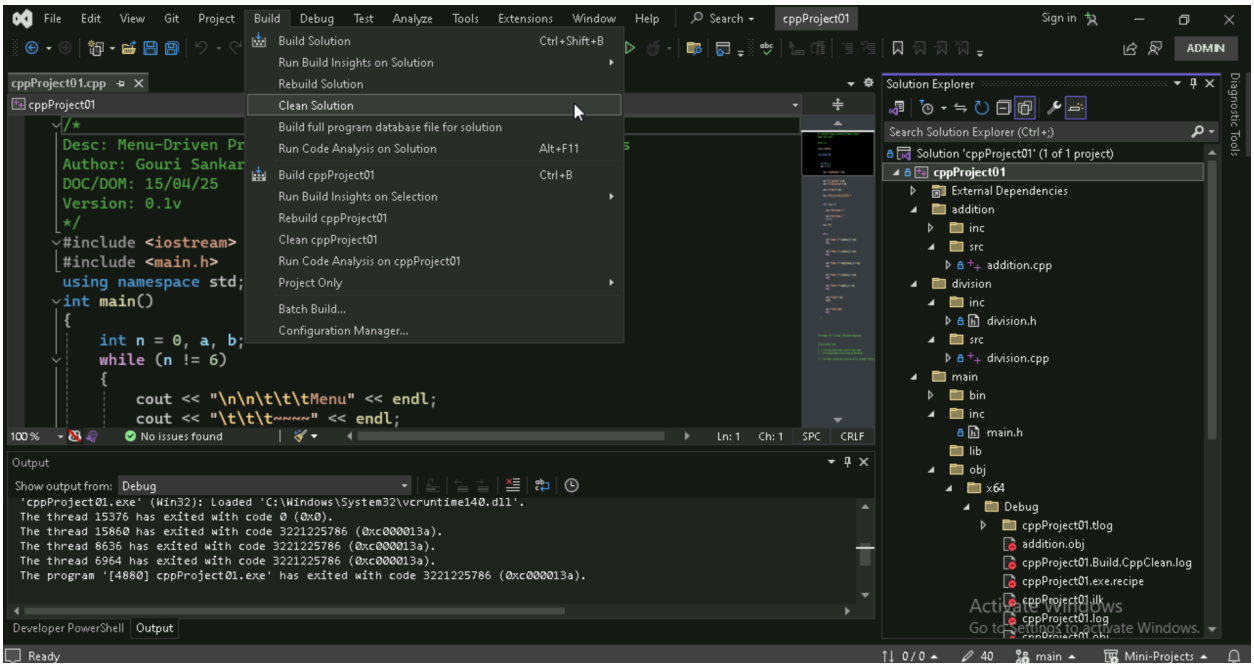
Deal with the Errors and Warnings

- Handle any errors and warnings that arise during the debugging
- Inorder to disable certain warnings, navigate to **Project Properties > C/C++ > Advanced > Disable Specific Warnings.**
- Add the ID numbers of the warnings to be disabled.

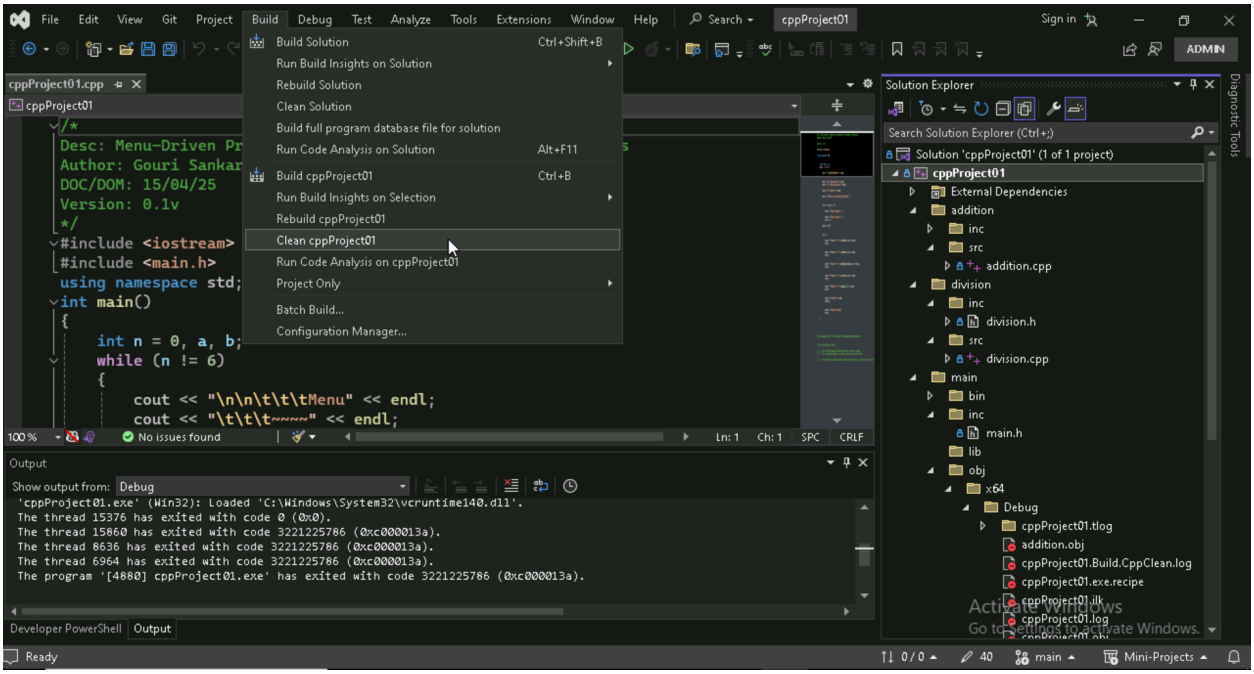


Build the application

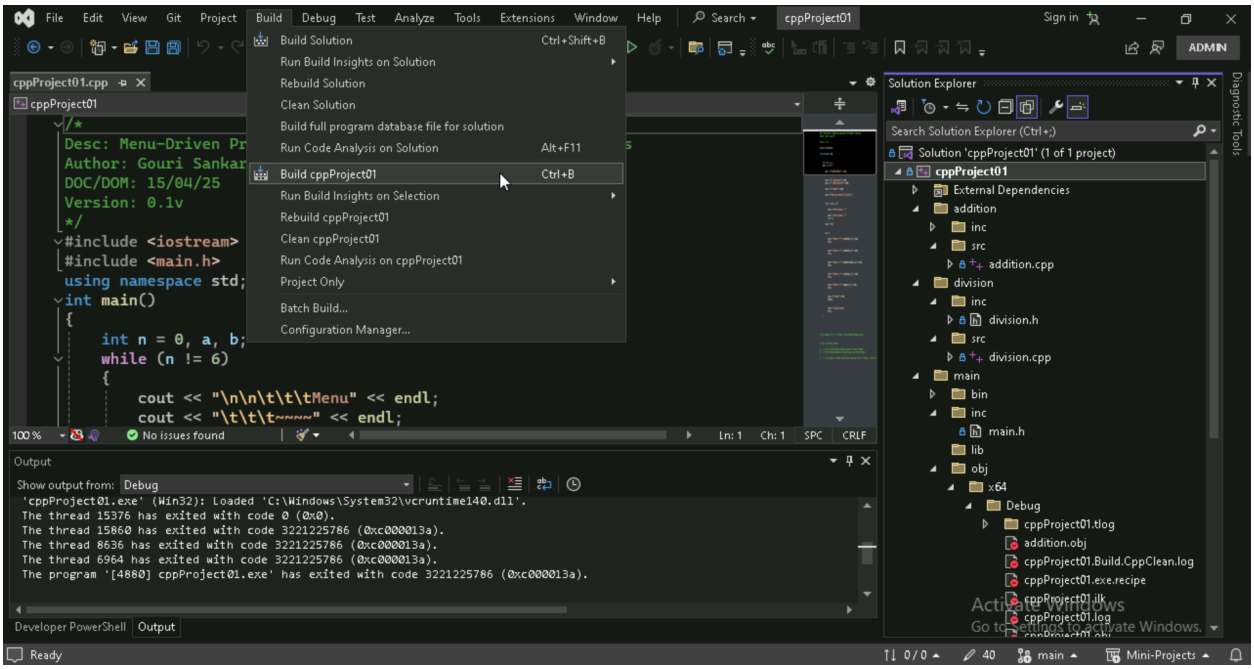
- Click on **Build > Clean Solution**



- Select **Build > Clean *'Project-Name'***



- Choose **Build > Build *'Project-Name'***


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
cppProject01
Sign in
ADMIN
Build Solution Ctrl+Shift+B
Run Build Insights on Solution
Rebuild Solution
Clean Solution
Build full program database file for solution
Run Code Analysis on Solution Alt+F11
Build cppProject01 Ctrl+B
Run Build Insights on Selection
Rebuild cppProject01
Clean cppProject01
Run Code Analysis on cppProject01
Project Only
Batch Build...
Configuration Manager...
cppProject01.cpp
cppProject01
/*
Desc: Menu-Driven Pr
Author: Gouri Sankar
DOC/DOM: 15/04/25
Version: 0.1v
*/
#include <iostream>
#include <main.h>
using namespace std;
int main()
{
int n = 0, a, b;
while (n != 6)
{
cout << "\n\n\t\t\tMenu" << endl;
cout << "\t\t\t~~~~" << endl;
100 %
No issues found
Ln: 1
Ch: 1
SPC
CRLF
Output
Show output from: Debug
'cppProject01.exe' (Win32): Loaded 'C:\Windows\System32\vcruntime140.dll'.
The thread 15376 has exited with code 0 (0x0).
The thread 15860 has exited with code 3221225786 (0xc000013a).
The thread 8636 has exited with code 3221225786 (0xc000013a).
The thread 6964 has exited with code 3221225786 (0xc000013a).
The program '[4880] cppProject01.exe' has exited with code 3221225786 (0xc000013a).
Developer PowerShell
Output
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'cppProject01' (1 of 1 project)
cppProject01
External Dependencies
addition
inc
src
addition.cpp
division
inc
division.h
src
division.cpp
main
bin
inc
main.h
lib
obj
x64
Debug
cppProject01.tlog
addition.obj
cppProject01.Build.CppClean.log
cppProject01.exe.recipe
cppProject01.ilk
cppProject01.log
Diagnostic Tools
Activate Windows
Go to Settings to activate Windows.
Ready
0 / 0
40
main
Mini-Projects