

## Things to note in Microsoft visual studio

### Project file formate :

To make our project with good readability and to maintainable well organised way . we can create project with structured folders and all files related to project , we can keep in the project folder according to the file type and structure.

Folder types and their content type :

#### 1) Source files folder ( src ) -

- Contains ".cpp" files , that implement functionalities of the project .

#### 2) Header files folder ( Include or headers or inc ) -

- Contains ".h" or ".hpp" files that declare functions, classes, and global variables.

#### 3) Intermediate files folder ( build or out ) -

- Contains ".o" or ".obj" files , compiled object files or intermediate files generated during the compilation process.
- Might include Makefiles as part of the build process.

#### 4) Binary folder ( bin ) -

- Holds the final executable.
- ".exe" , the finale excutable file generated by linking all .obj files

#### 5) External libraries folder ( lib or ext ) -

- Shared libraries ".so" ( linux ) or ".dll" ( windows).
- static libraries ".a" or ".a" ( linux ) or ".lib" ( windows).

#### 6) Assets folder ( assets ) -

- Contains resources like images, icon, or other files required for the project.

#### 7) Test folder ( tests ) -

- Contains unit tests and related testing files.
- Test scripts written in ".cpp" or other languages , scripts/program to test our project with test cases.
- Result logs stored in ".log" or ".txt"

#### 8) Documentation folder ( docs ) -

- Contains documentation like README files or technical guides in formats such as ".md" or ".txt".

#### 9) Configuration folder ( config or configs ) -

- Contains configuration files like ".json" , ".xml" , ".yaml" or ".int" , to project setup, such as settings for libraries , compiler flags , or environment variables.

eg:

- src -> main.cpp , add.cpp
- inc -> add.h
- lib -> libadd.so
- build -> \*.o
- bin -> "main.exe" or "app.exe" or "app"
- tests -> test\_add.cpp ( test cases checking script for the add function
- docs -> README.md ( project documentation )

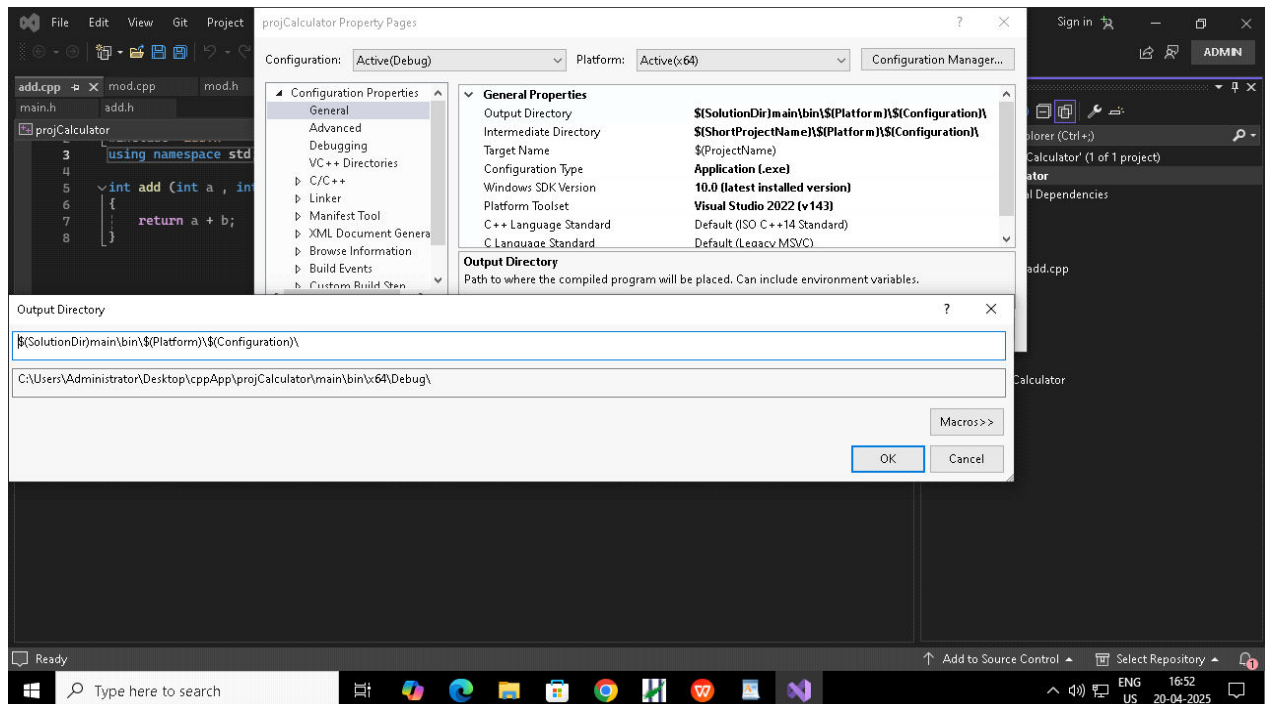
- Makefile -> automates the building .

### Bin path :

To change the bin path ,

Project\_properties > Configuration\_properties > General > Output\_directory > click\_on\_it > click\_on\_Down\_arrow\_symbol\_on\_it > edit > Give \_the\_required\_path > Click\_ok.

Setting path in IDE to store the generated ".exe" in our target bin folder .

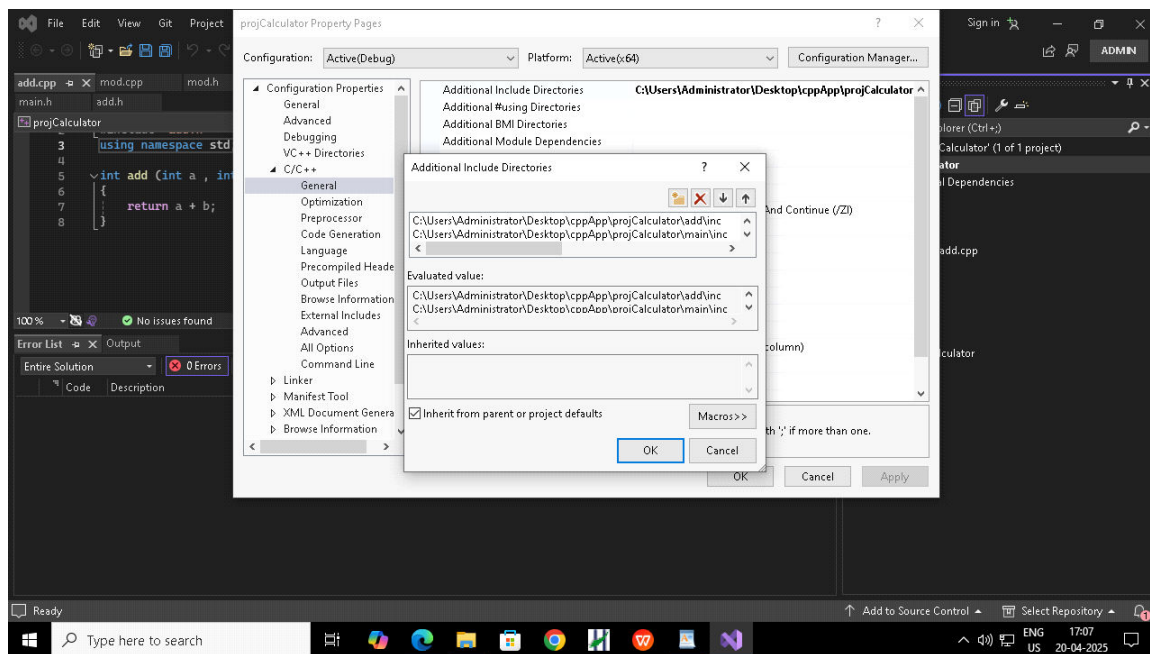


## Inc path

To add include paths (header file).

Project\_properties > C/C++ > General > Additional\_include\_directories > click\_on\_it > click\_on\_Down\_arrow\_symbol\_on\_it > edit > Give \_the\_required\_path > Click\_ok.

In preprocessing stage the IDE should know the path of the header file , so we should give the path of headers which we create in our defined location.



## Obj path :

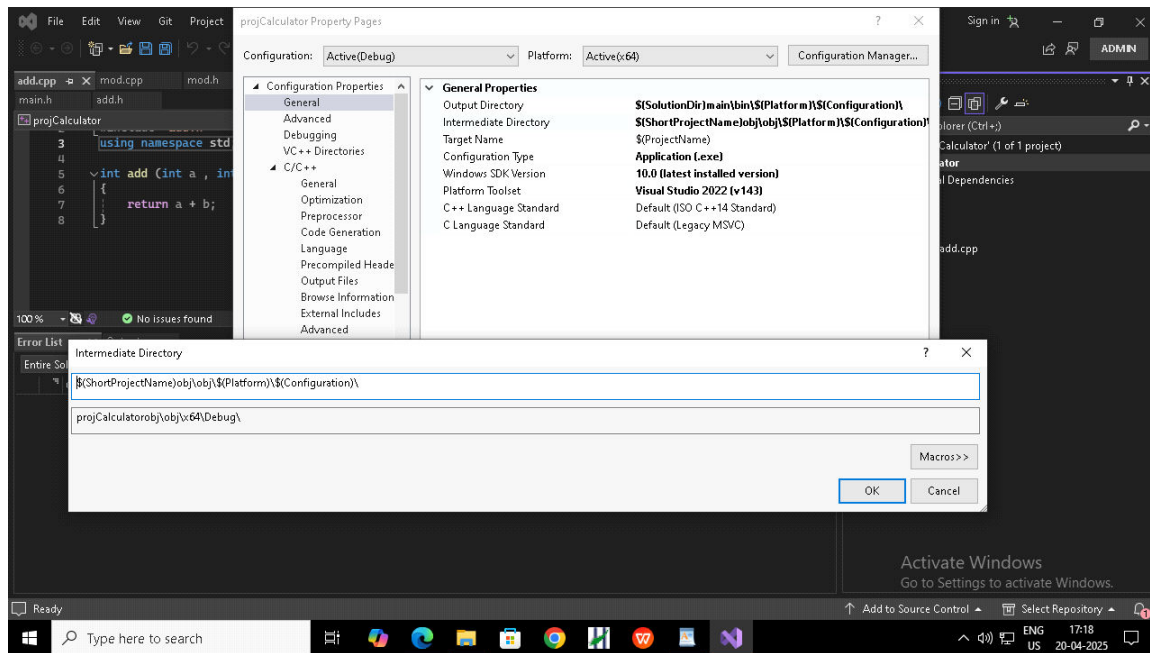
To store the intermediate file in our required location , we can

change the path location.

Project\_properties > Configuration\_properties > General >

Intermediate\_directory > click\_on\_it >

click\_on\_Down\_arrow\_symbol\_on\_it > edit > Give \_the\_required\_path  
> Click\_ok.



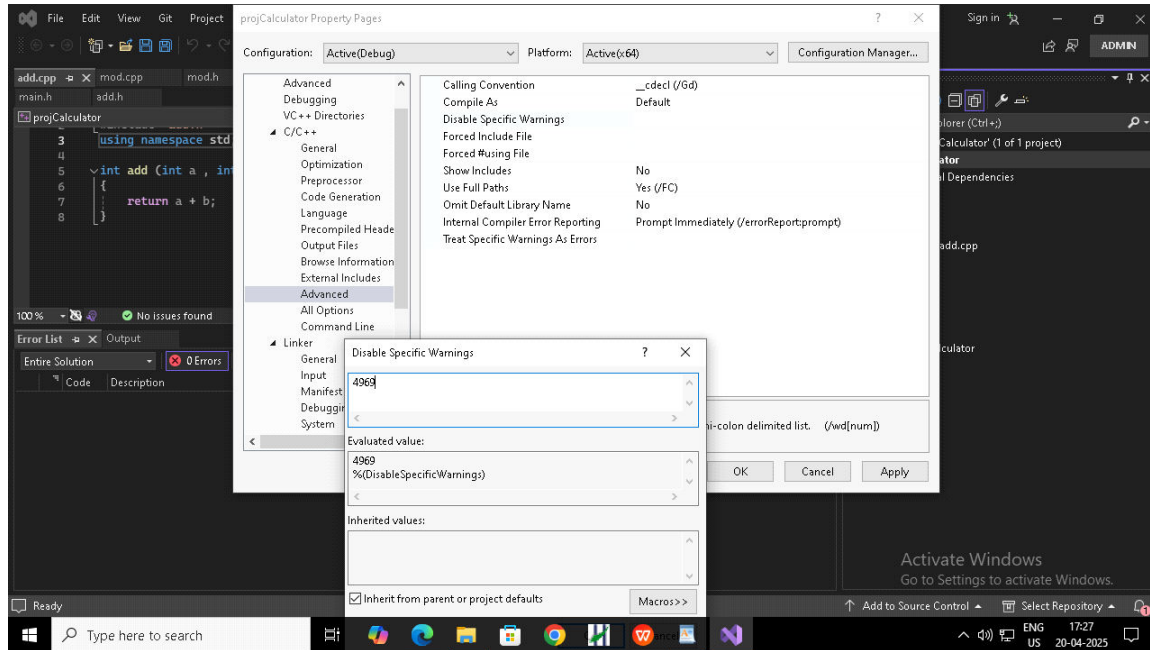
### Warning compromise :

To compromise the list of warnings .

method1 ( using warning code ) :

Project\_properties > C/C++ > Advanced > Disable\_specific\_warnings >

click\_on\_it > click\_on\_Down\_arrow\_symbol\_on\_it > edit > Give  
\_the\_warning\_code > Click\_ok.



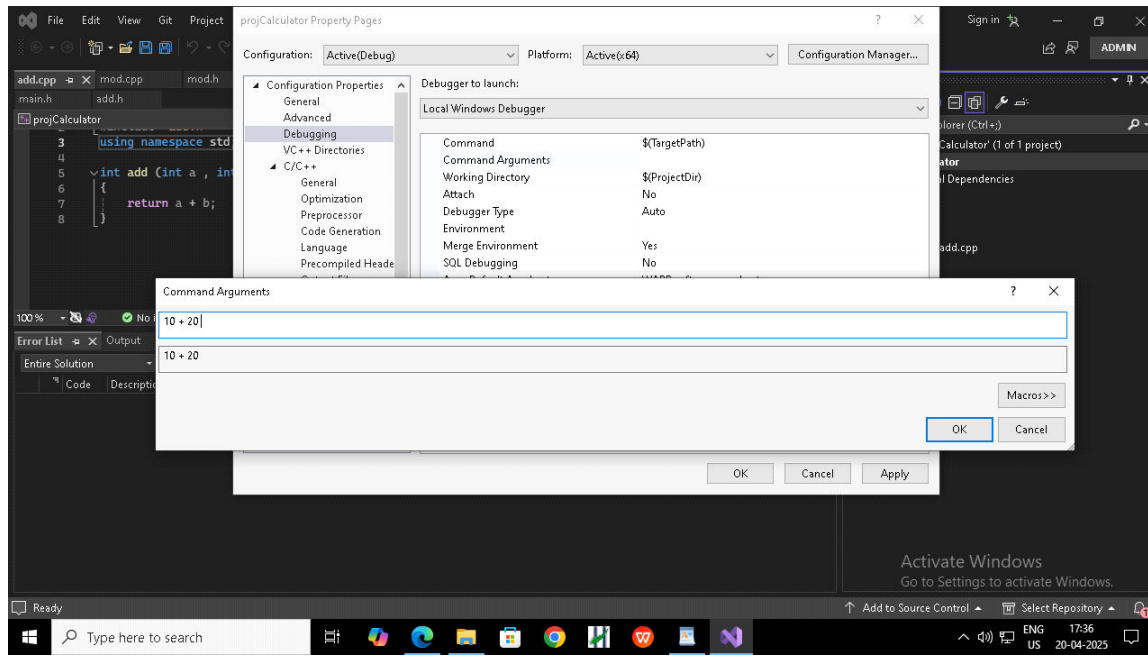
method2 ( by using macro of that warning ) :

Project\_properties > C/C++ > Preprocessor >  
 Preprocessor\_definitions > click\_on\_it >  
 click\_on\_Down\_arrow\_symbol\_on\_it > edit > Give \_the\_warning\_code  
 > Click\_ok.

## Command line Arguments

To give command line arguments to the program , we give the command line arguments in that location , then IDE will feed that CLAs when the program is starting to excute .

Project\_properties > Configuration\_properties > Debugging >  
 Command\_arguments > click\_on\_it >  
 click\_on\_Down\_arrow\_symbol\_on\_it > edit > Give \_the\_warning\_code  
 > Click\_ok.



## Exclude from compilation :

To exclude a file from compilation ,

Select\_the\_file > Right\_click > properties > Configuration\_properties > General > Exclude from build .

