C++ Computer Science

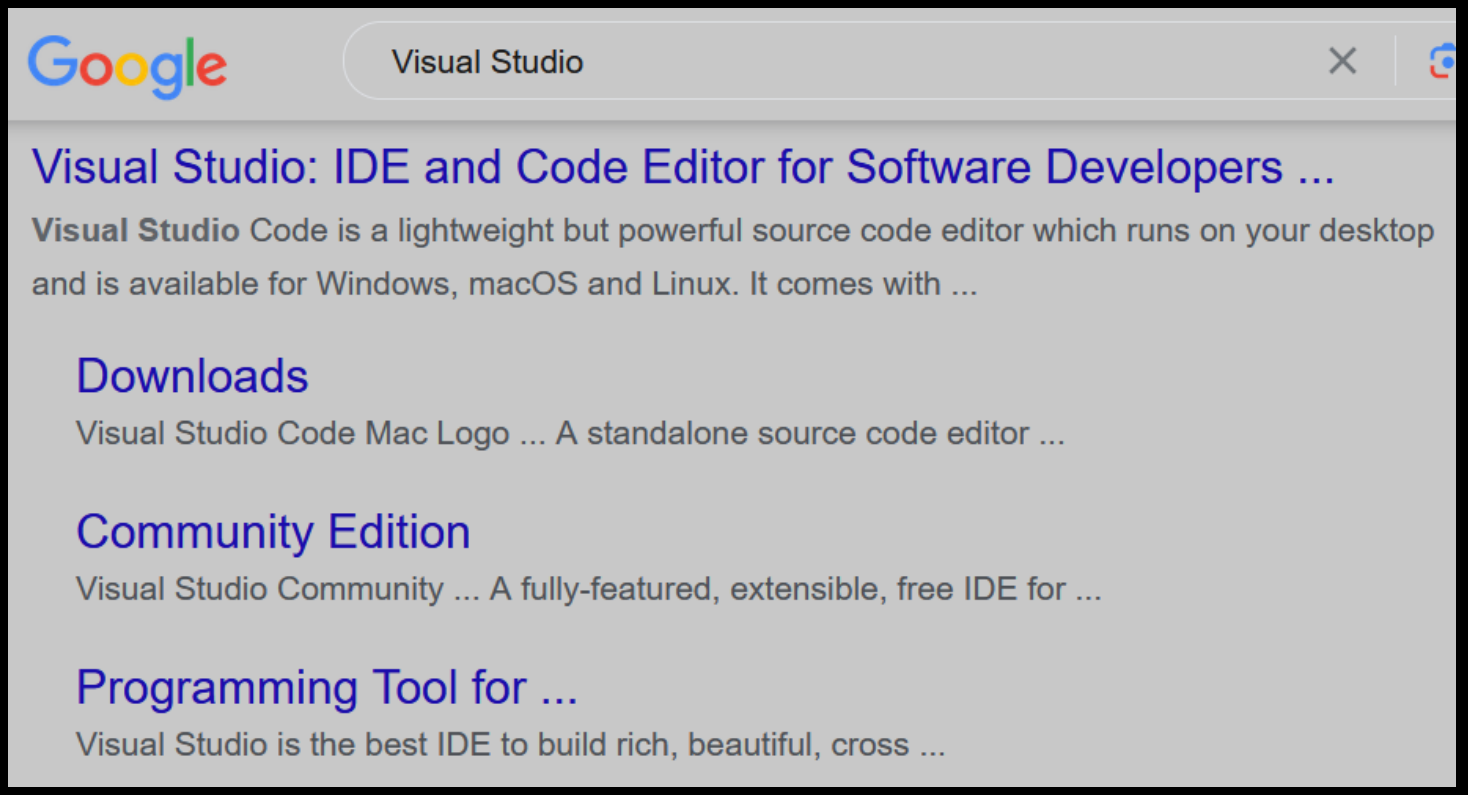
by

Christopher Andrew Topalian

Christopher Topalian

Dedicated to God the Father

Download Visual Studio - Search Google



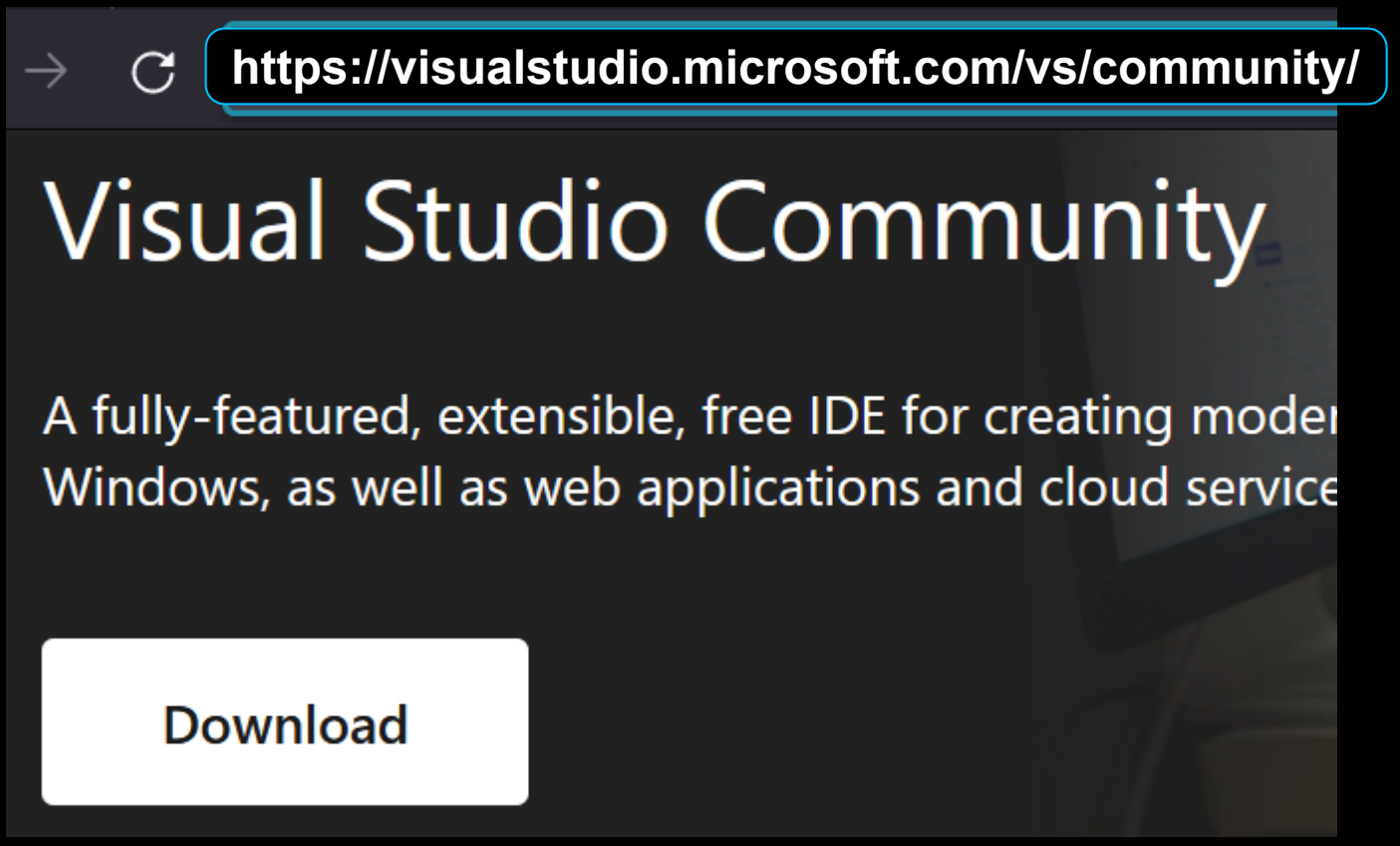
We Go To: google.com

We Search for: Visual Studio

We Left Click on: Downloads

Or we can go directly
to the Visual Studio website
as shown on the next page.

Download Visual Studio - Directly from Website



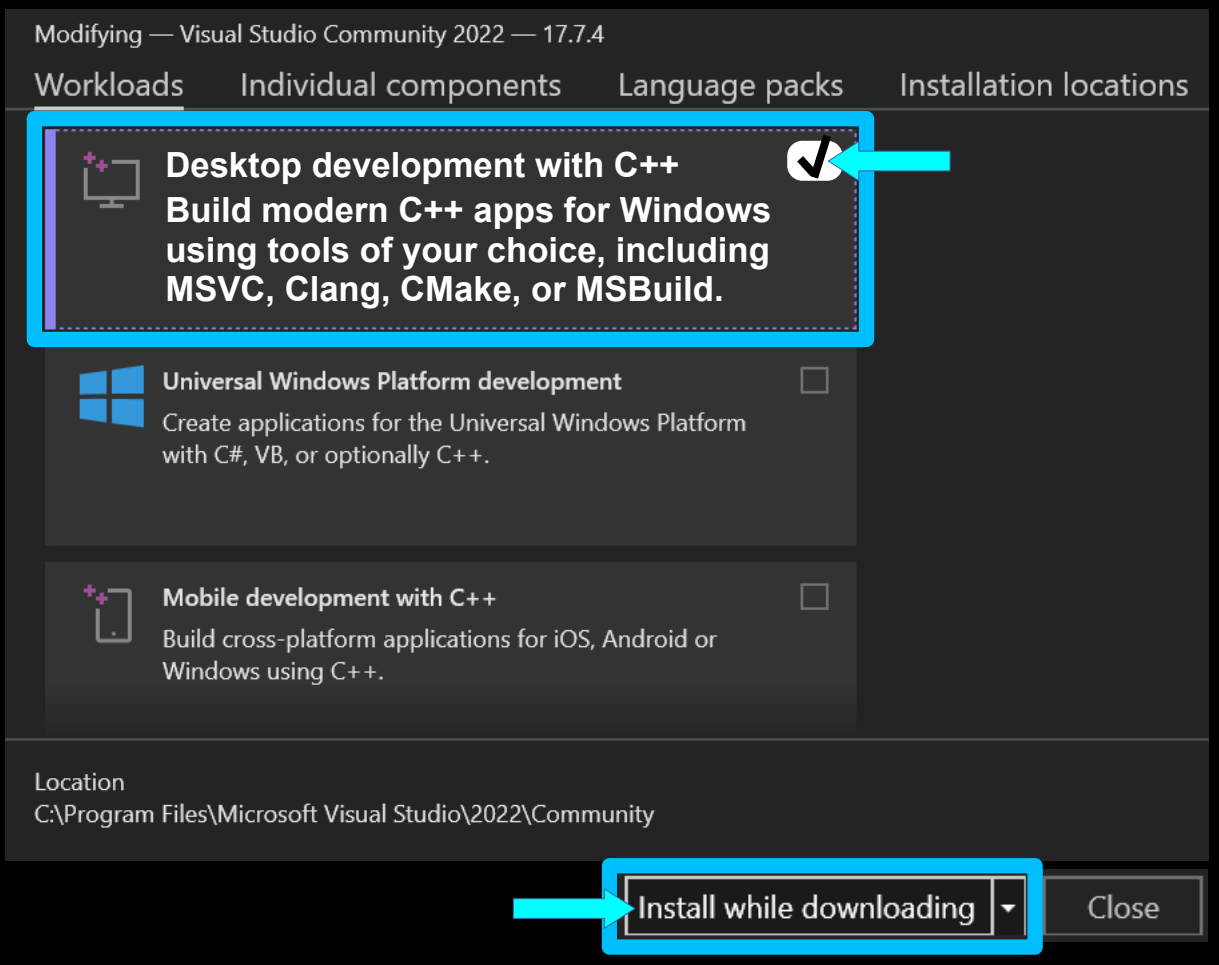
We Go To:
https://visualstudio.microsoft.com/vs/community

We Download: **Visual Studio Installer**

We Go To our Downloads Folder and:
Double Left Click the Install file to Install it.

After it is installed, we can open VS Studio.
Once, open, we can then install the C++ package.

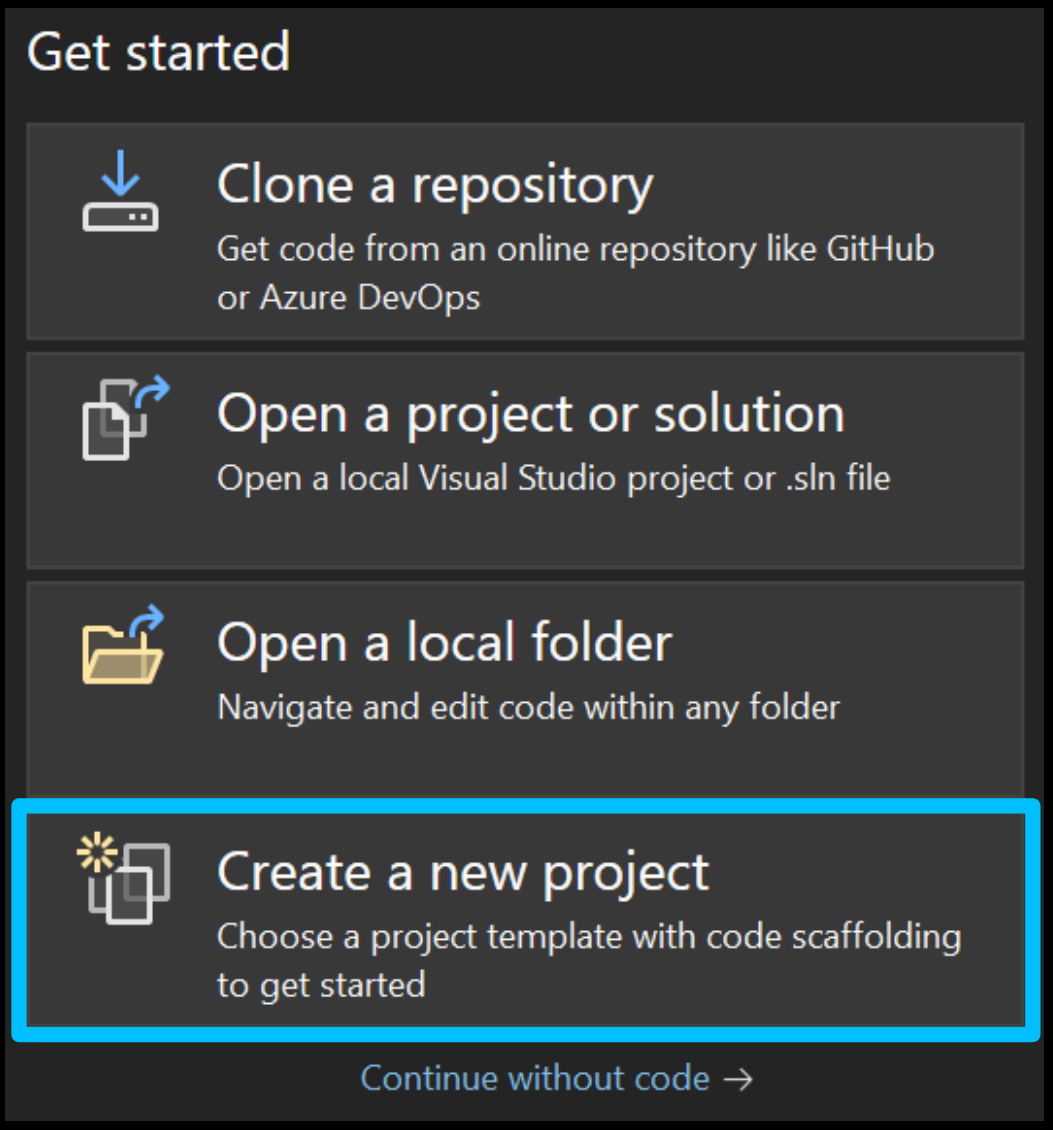
We Download and Install:
Desktop development with C++



We Put a Checkmark in the box and then Left Click the Install while downloading button

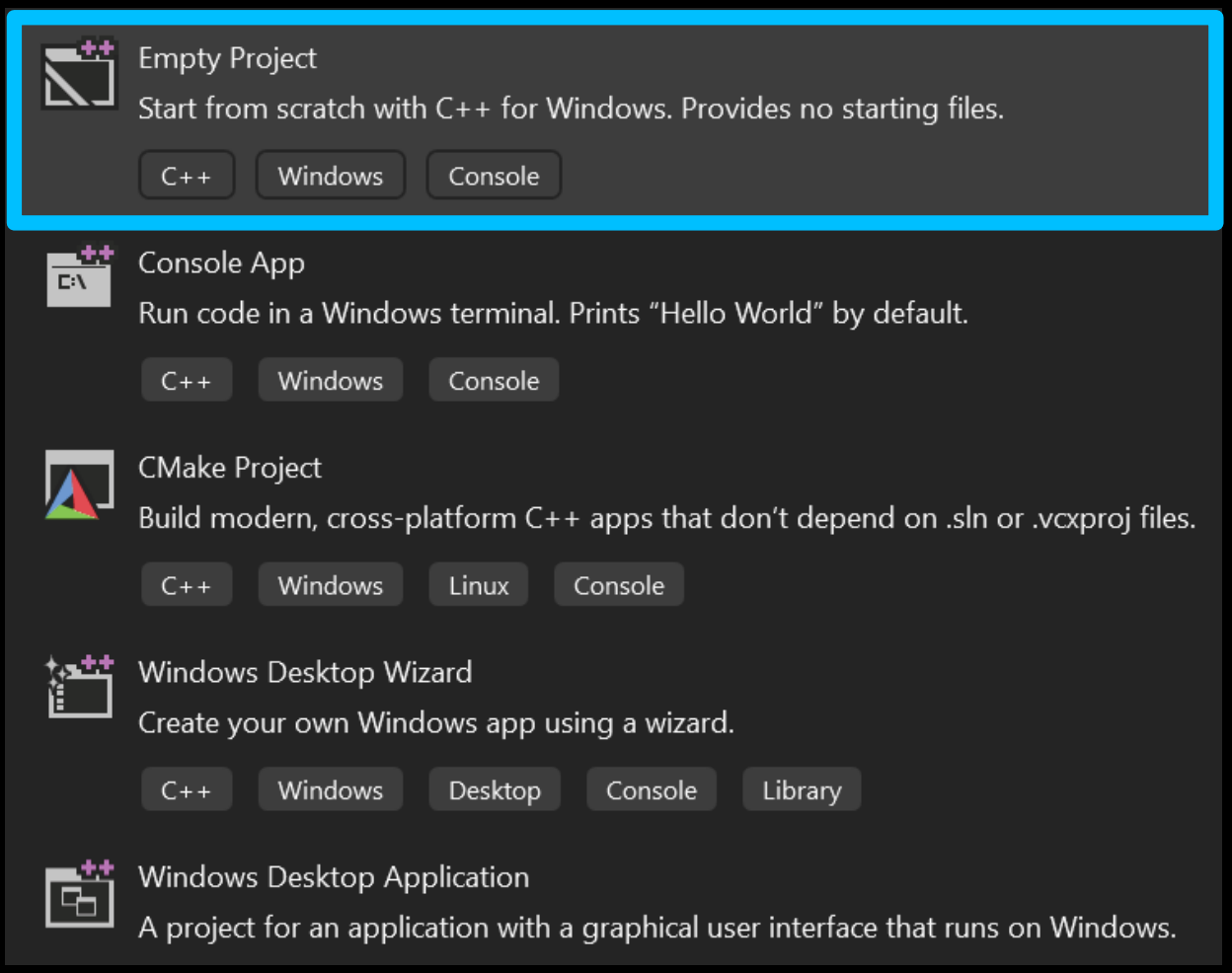
This will download and install the ability to use Visual Studio to create C++ Desktop Applications.

Create a New Project

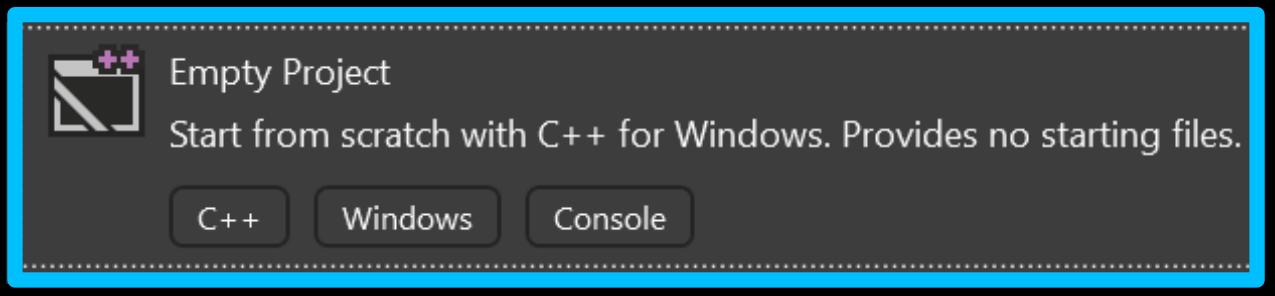


We Choose: Create a new project

Choices for a New Project

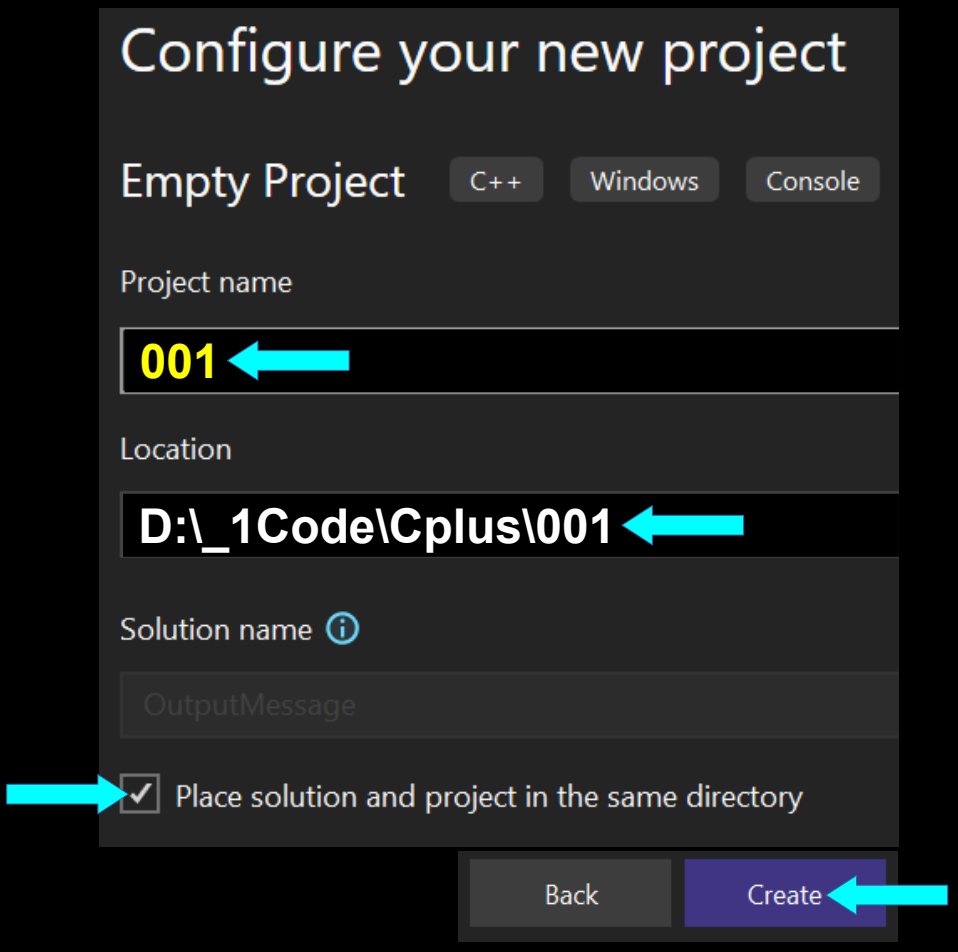


We Left Click: Empty Project



We Left Click: Next Button

Project Name - 001



We name our first project as: 001

We put a Checkmark in: Place solution and project in the same directory

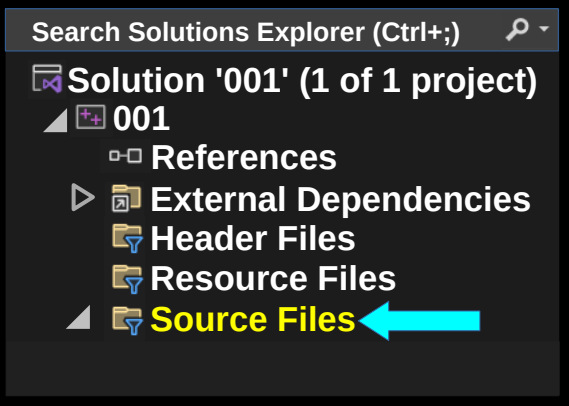
We Left Click on: Create Button

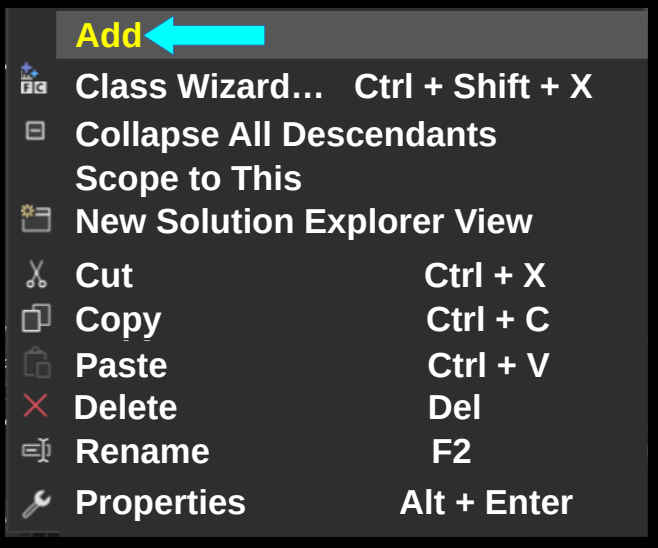
Creating our main.cpp file in Source Files Folder

We Right click on: Source Files Folder

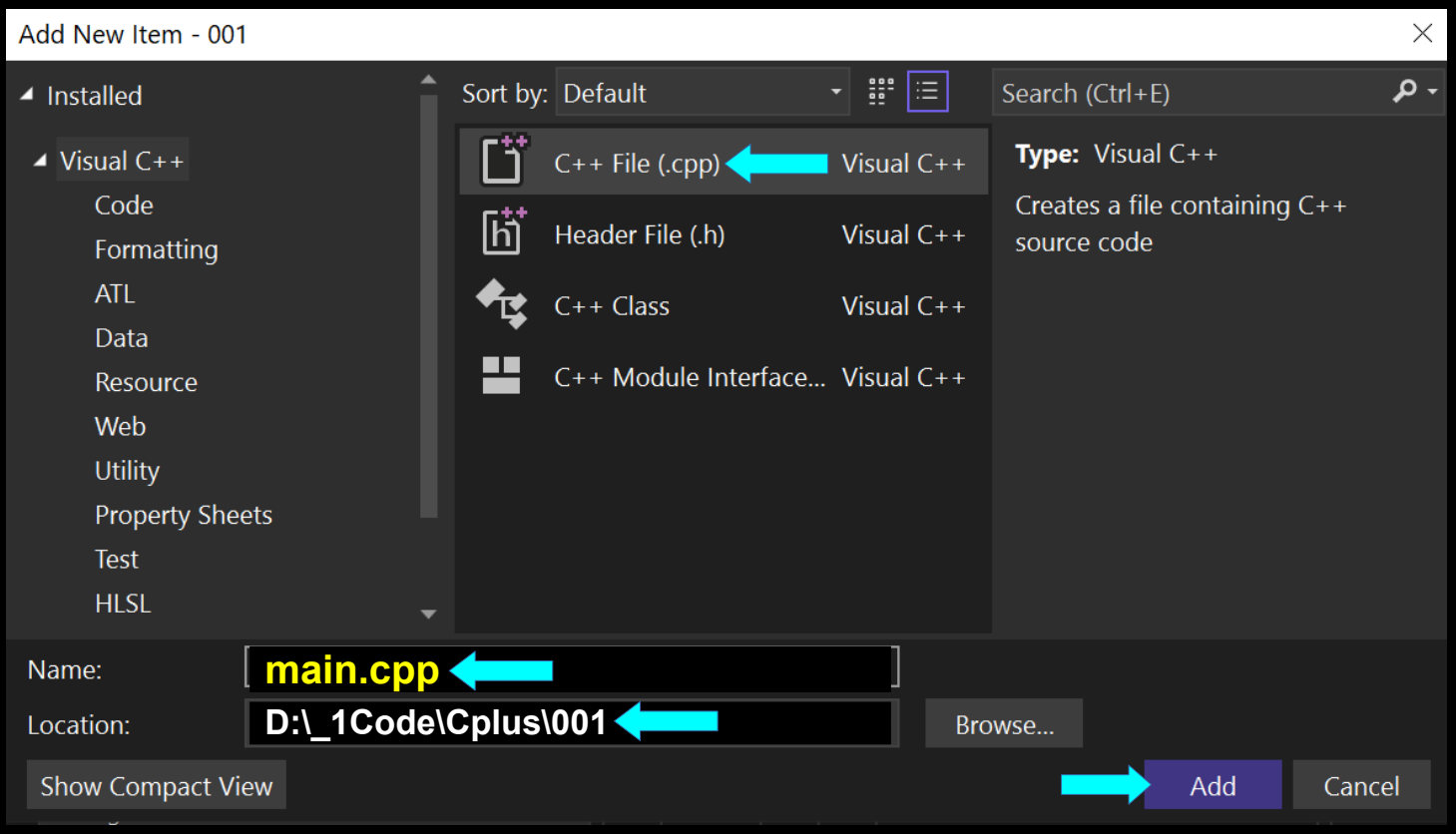
We Choose: Add

We Choose: New Item...





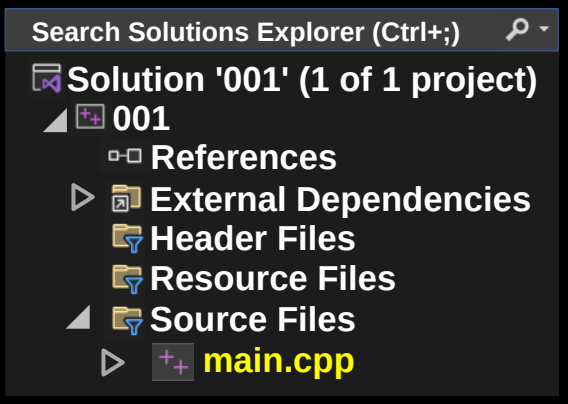
We Choose: C++ File (.cpp)



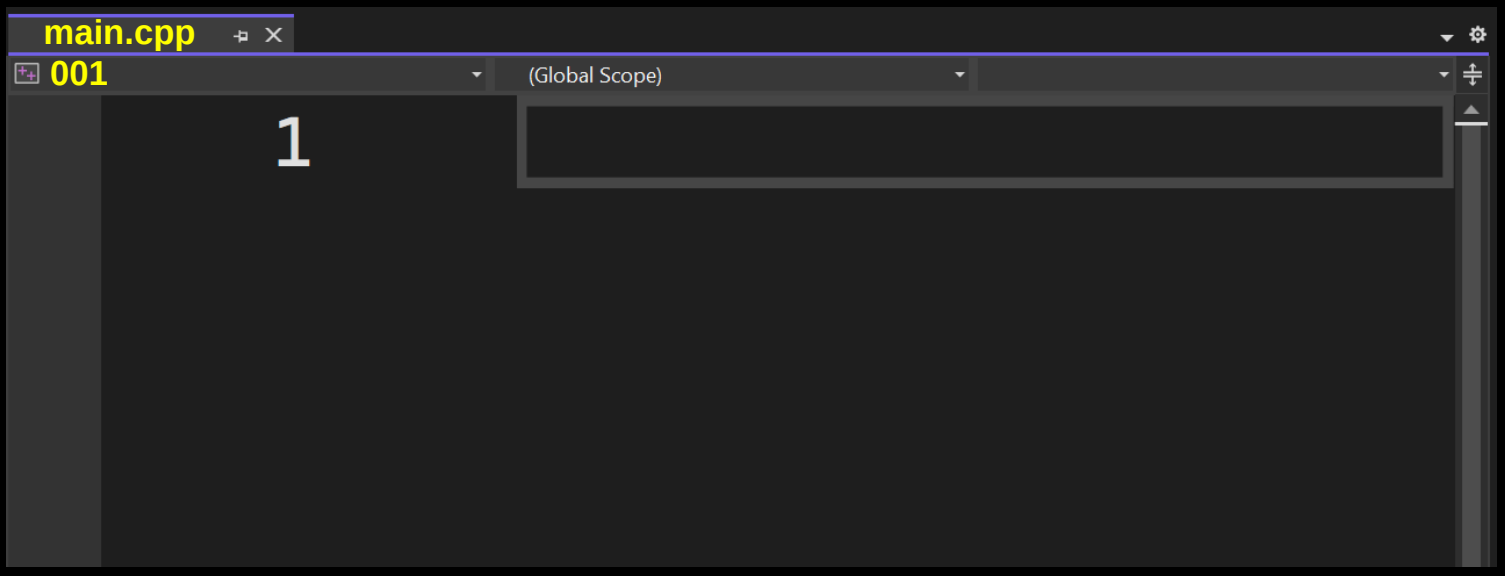
We name our file: main.cpp

We Left Click: Add button

We see our created file: main.cpp



main.cpp is now open



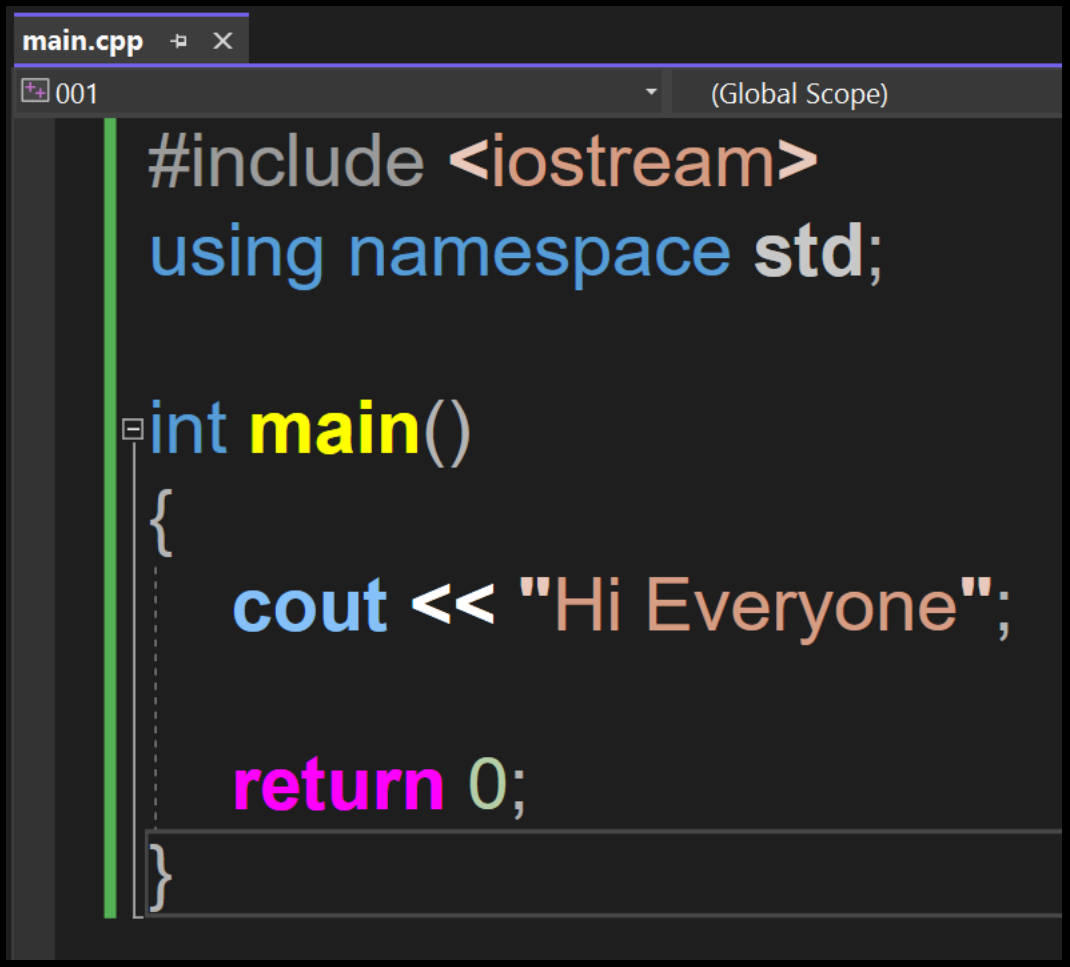
We can now: Type our Code :-)

We make Bigger Font by:
Control + Scroll Wheel Forward

We make Smaller Font by:
Control + Scroll Wheel Backward

main.cpp Code - Screenshot

Here is a screenshot of: Our C++ Code

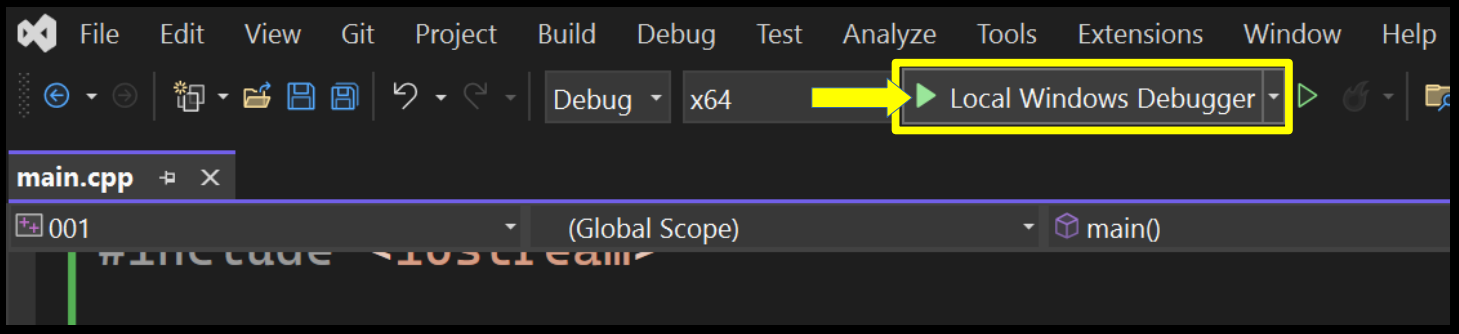


On the next page
we show the same code,
but with better font.

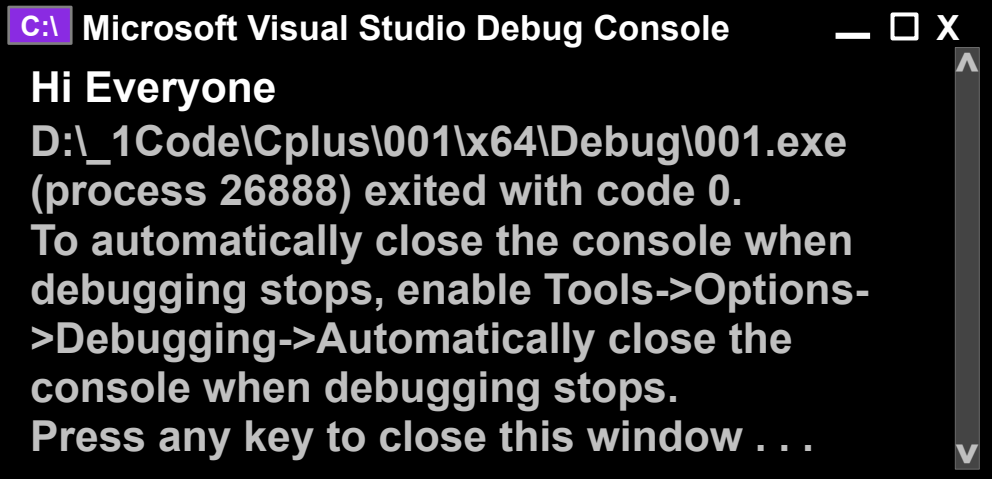
```
// Outputting Text

// main.cpp

#include <iostream>
using namespace std;

int main()
{
    cout << "Hi Everyone";

    return 0;
}
```

// Building and Running our App

We Left Click on: Local Windows Debugger



Our app opens in the Debug Console Window, with the message of: Hi Everyone



We make Bigger Console Font by:
Control + Scroll Wheel Forward

We make Smaller Console Font by:
Control + Scroll Wheel Backward

```
// Outputting Text and Exit by Pressing Enter

// main.cpp

#include <iostream>
using namespace std;

int main()
{
    cout << "Hi Everyone" << "\n";

    cout << "Press Enter to Exit";
    cin.get();

    return 0;
}
```

```
// Input from user

// main.cpp

#include <iostream>
#include <string>
using namespace std;

int main()
{
    string name;

    cout << "Enter Name: ";
    cin >> name;

    cout << "Hi " << name;

    cout << "\nPress Enter to Exit";
    cin.ignore();
    cin.get();

    return 0;
}
```

```
// Custom function - askName

// main.cpp

#include <iostream>
#include <string>
using namespace std;

string askName()
{
    string name;
    cout << "Enter Name: ";
    cin >> name;
    return name;
}

int main()
{
    string userName = askName();
    cout << "Hi, " << userName << "!" << "\n";

    cout << "\nPress Enter to Exit";
    cin.ignore();
    cin.get();
    return 0;
}
```

C:\ D:_1Code\Cplus\001\x64\Debug\001.exe

```
Enter Name: Christopher
Hi, Christopher!

Press Enter to Exit
```

```
// Custom Function - consoleLog

// main.cpp

#include <iostream>
#include <string>
using namespace std;

void consoleLog(const string message)
{
    cout << message << "\n";
}

int main()
{
    consoleLog("Hi Everyone");

    cout << "Press Enter to Exit";
    cin.get();

    return 0;
}
```

Header File - We define our function in a header file for easy use

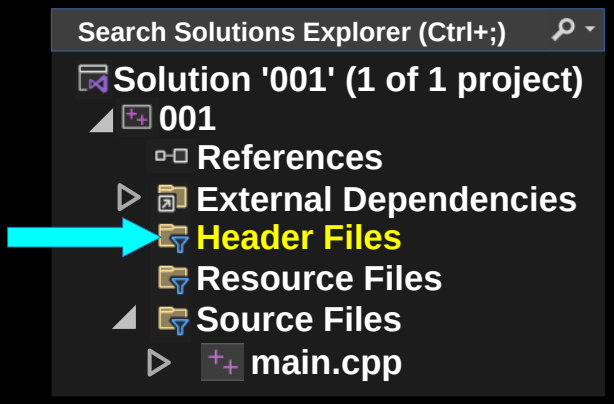
Instead of pasting this useful function in every script in our application, we will instead type it once in a header file and put it in the Header Files Folder.

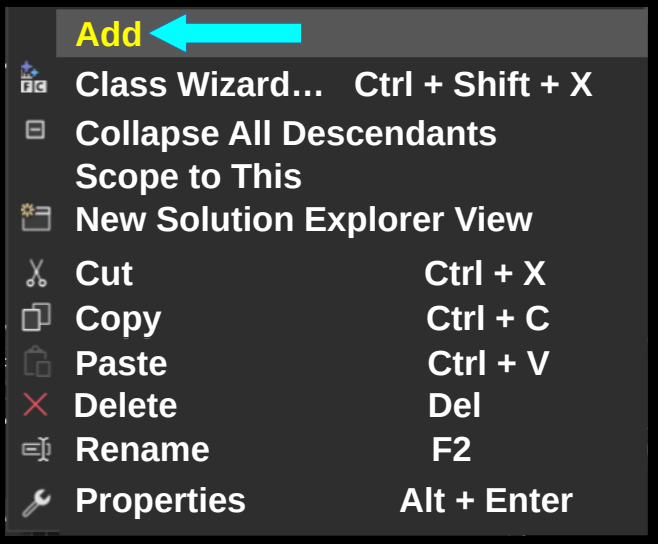
Using a header file is easier, because we place the header files in the Header Files folder and then include that header file with a reference in our main.cpp and other files.

In our header file, we type the terms
#ifndef
and
#define
to designate that it will be used in other files.

Header File - Add - New Item

We right click on: Header Files folder

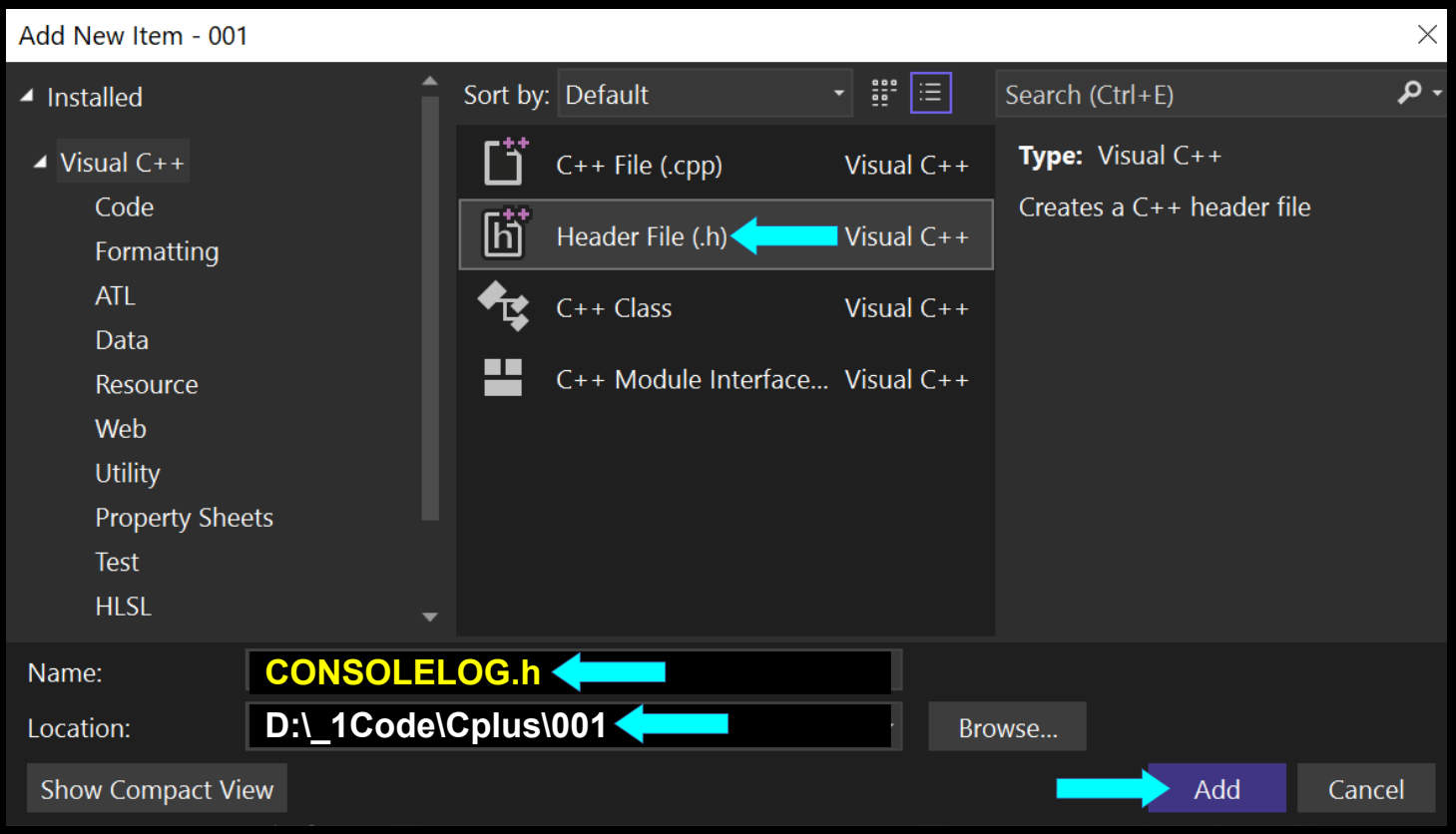




We choose: Add

We choose: New Item

We choose: Header File (.h)



We name it: CONSOLELOG.h

We Left Click: Add button

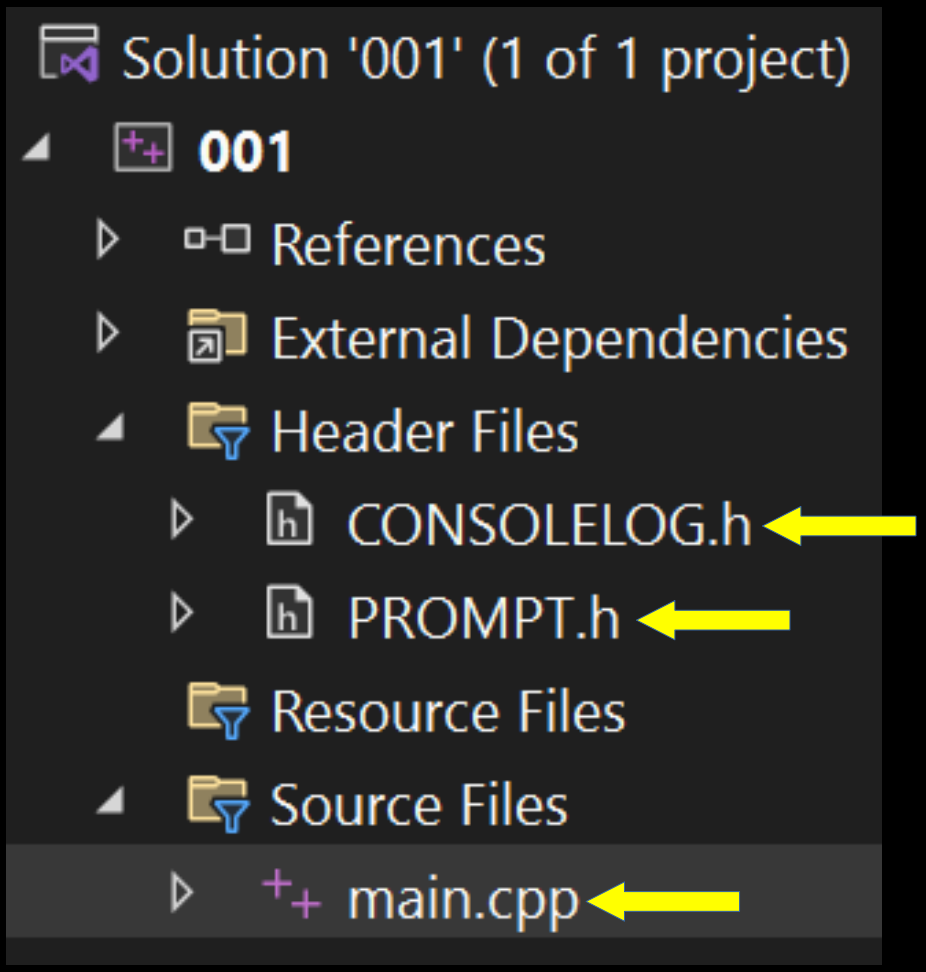
```
// CONSOLELOG.h header file

// CONSOLELOG.h

#ifndef CONSOLELOG
#define CONSOLELOG
#include <iostream>
#include <string>

void consoleLog(const std::string message)
{
    std::cout << message << "\n";
}

#endif
```

```
// Our main.cpp file uses the CONSOLE.h file

// main.cpp

#include <iostream>
#include <string>
#include "CONSOLELOG.h"
using namespace std;

int main()
{
    consoleLog("Hi Everyone");

    cout << "Press Enter to Exit";
    cin.get();

    return 0;
}
```

```
// PROMPT.h header file

// PROMPT.h

#ifndef PROMPT
#define PROMPT
#include <iostream>
#include <string>

void prompt(std::string& userInput)
{
   std::cin >> userInput;
}

#endif
```

```
// main.cpp uses CONSOLELOG.h and
PROMPT.h

// main.cpp


#include <iostream>
#include <string>
#include "PROMPT.h"
#include "CONSOLELOG.h"
using namespace std;

int main()
{
    string input;

    consoleLog("Enter Name");
    prompt(input);

    consoleLog("Hi " + input);

    consoleLog("Press Enter to Exit");
    cin.ignore();
    cin.get();

    return 0;
}
```

File Structure of the previous Examples



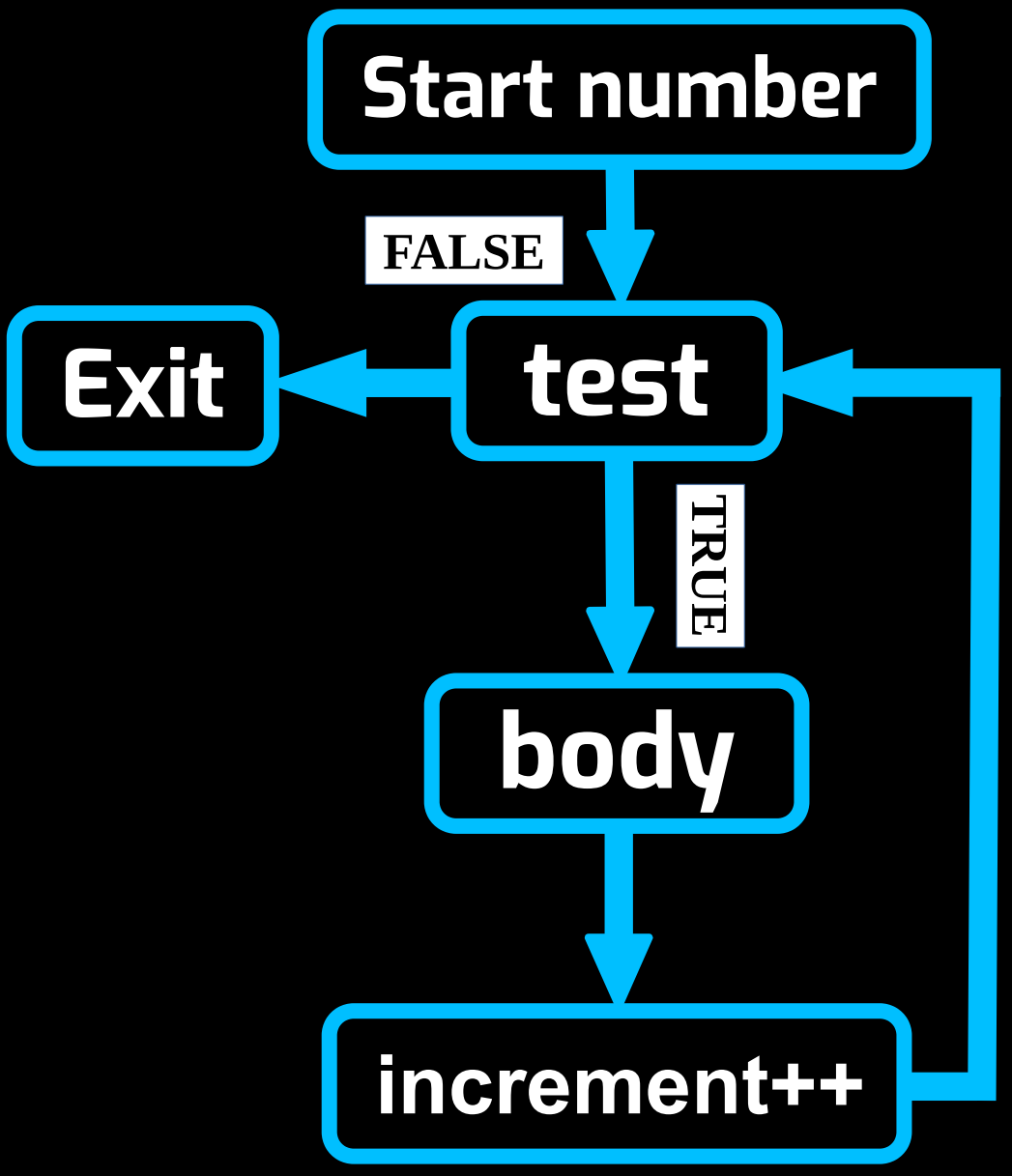
We have 2 Header Files:
CONSOLELOG.h
and
PROMPT.h

We have 1 main.cpp file:
main.cpp uses CONSOLELOG.h and PROMPT.h header files

```
// Array of Objects

// main.cpp

#include <iostream>
#include <string>
#include <vector>
using namespace std;

struct Person
{
    string name;
    int age;
};

int main()
{
    vector<Person> people =
    {
        {
            "Alice",
            25
        },

        {
```

```
        "Bob",
        30
    },

    {
        "Jane",
         28
    }
};

for (const auto person : people)
{
    cout << "Name: "
    << person.name
    << ", Age: "
    << person.age
    << "\n";
}

cout << "\nPress Enter to Exit";
cin.get();

return 0;
}
```

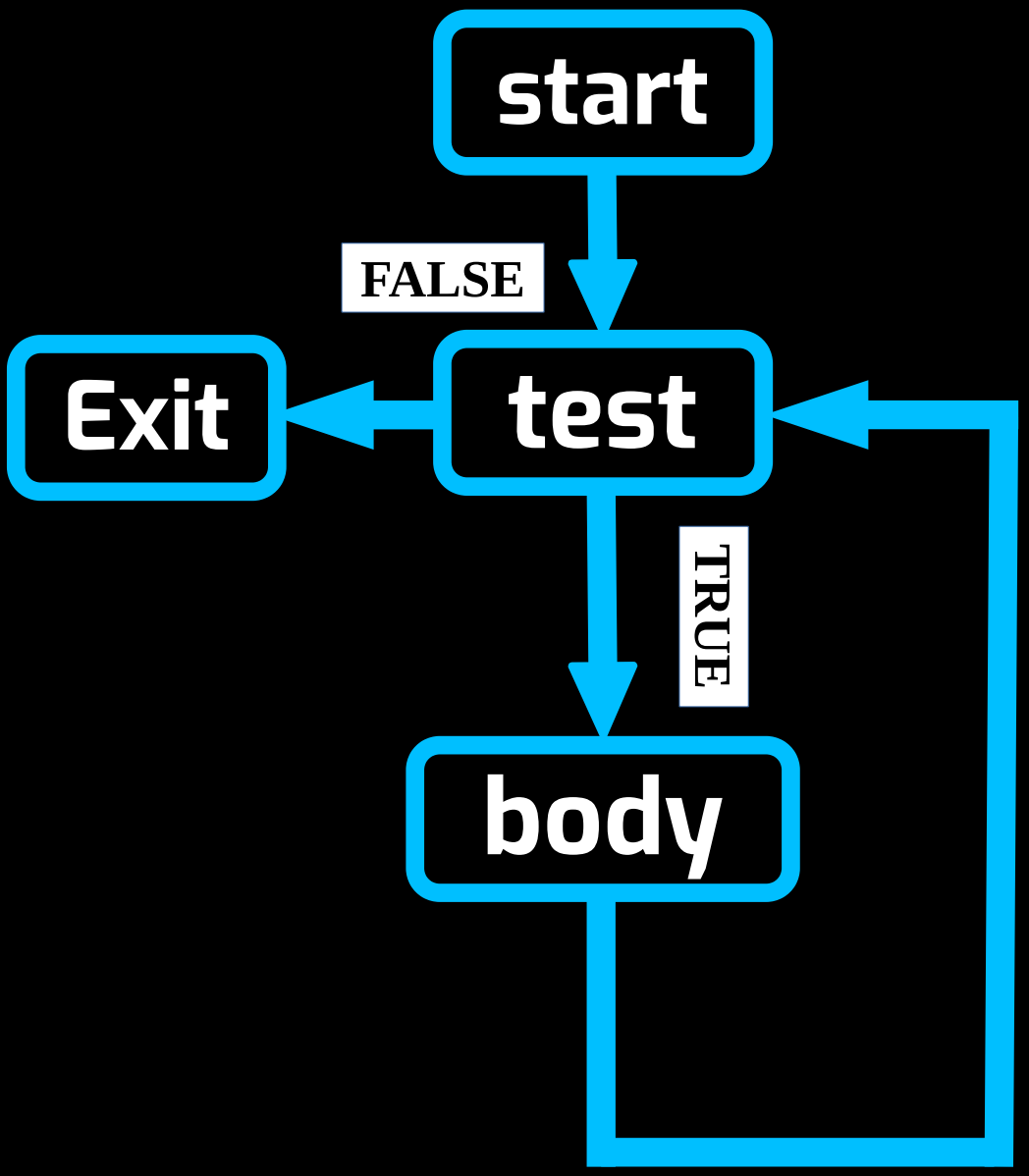
// for loop diagram



```
// for loop

// main.cpp

#include <iostream>
using namespace std;

int main()
{
    for (int i = 1; i <= 100; i++)
    {
        cout << i << " ";
        cout << "\n";
    }

    cout << "\nPress Enter to Exit";
    cin.get();

    return 0;
}
```

Christopher Topalian

// while loop diagram



```
// while loop

// main.cpp

#include <iostream>
using namespace std;

int main()
{
    int count = 0;

    while (count < 5)
    {
        cout << "Count: "
        << count + 1
        << "\n";
        count++;
    }

    cout << "Loop completed." << "\n";
    cout << "\nPress Enter to Exit";
    cin.get();

    return 0;
}
```

```
// if else

#include <iostream>
#include <string>
using namespace std;

// main.cpp

int main()
{
    string name;
    cout << "Enter your name: ";
    cin >> name;

    if (name == "Chris")
    {
        cout << "Hi Chris.\nIt is good that you are
visiting Earth." << "\n";
    }
    else
    {
        cout << "Howdy " << name << ". "
        << "Tell Chris to Sign in later."
        << "\n";
    }
```

```
    cout << "\nPress Enter to Exit";
    cin.ignore();
    cin.get();

    return 0;
}
```

```
C:\ D:\_1Code\Cplus\001\x64\Debug\001.exe
Enter your name: John
Howdy John. Tell Chris to Sign in later.

Press Enter to Exit_
```

```
C:\ D:\_1Code\Cplus\001\x64\Debug\001.exe
Enter your name: Chris
Hi Chris.
It is good that you are visiting Earth.

Press Enter to Exit_
```

```
// Open Browser to a URL

// main.cpp

#include <windows.h>

int main()
{
    ShellExecuteA(NULL, "open",
"https://www.google.com", NULL, NULL,
SW_SHOWNORMAL);

    return 0;
}
```

```
// Custom Function - Open Browser to a URL

// main.cpp

#include <windows.h>
#include <string>
using namespace std;

void openURL(const string url)
{
    ShellExecuteA(NULL, "open", url.c_str(),
NULL, NULL, SW_SHOWNORMAL);
}

int main()
{
    string url = "https://www.google.com";

    openURL(url);

    return 0;
}
```

```
// Create Text File with Data

// main.cpp

#include <iostream>
#include <fstream>
using namespace std;

int main()
{
    // open file for writing
    std::ofstream outputFile("ourTextFile.txt");

    // write data to file
    outputFile << "Hi Everyone" << "\n";

    // close file
    outputFile.close();

    cout << "Data written successfully."
    << "\n";

    return 0;
}
```

```
// Custom Function - Create Text File with Data

// main.cpp

#include <iostream>
#include <string>
#include <fstream>
using namespace std;

void writeToFile(const string filename, const
string content)
{
    // open file for writing
    std::ofstream outputFile(filename);

    // write data to the file
    outputFile << content << "\n";

    // close the file
    outputFile.close();

    cout << "Data written to "
    << filename << " successfully."
    << "\n";
}
```

```
int main()
{
    string filename = "ourTextFile.txt";

    string content = "Hi Everyone";

    writeToFile(filename, content);

    return 0;
}
```

```
// Read a Text File

// main.cpp

#include <iostream>
#include <string>
#include <fstream>
using namespace std;

void displayFileContents(const string filename)
{
    // open file for reading
    std::ifstream inputFile(filename);

    // check if the file is open
    if (!inputFile.is_open())
    {
        std::cerr << "Error opening file: "
        << filename
        << "\n";

        return;
    }

    // read and display file contents
```

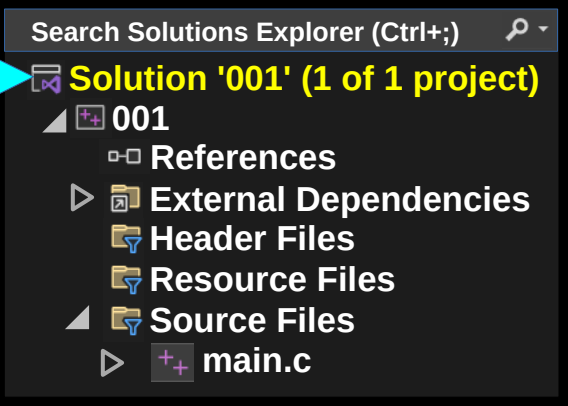
```
    string line;

    cout << "Contents of "
    << filename << ":" << "\n";

    while (std::getline(inputFile, line))
    {
        cout << line << "\n";
    }

    // close the file
    inputFile.close();
}

int main()
{
    string filename = "ourTextFile.txt";

    displayFileContents(filename);

    cout << "\nPress Enter to Exit";
    cin.get();

    return 0;
}
```

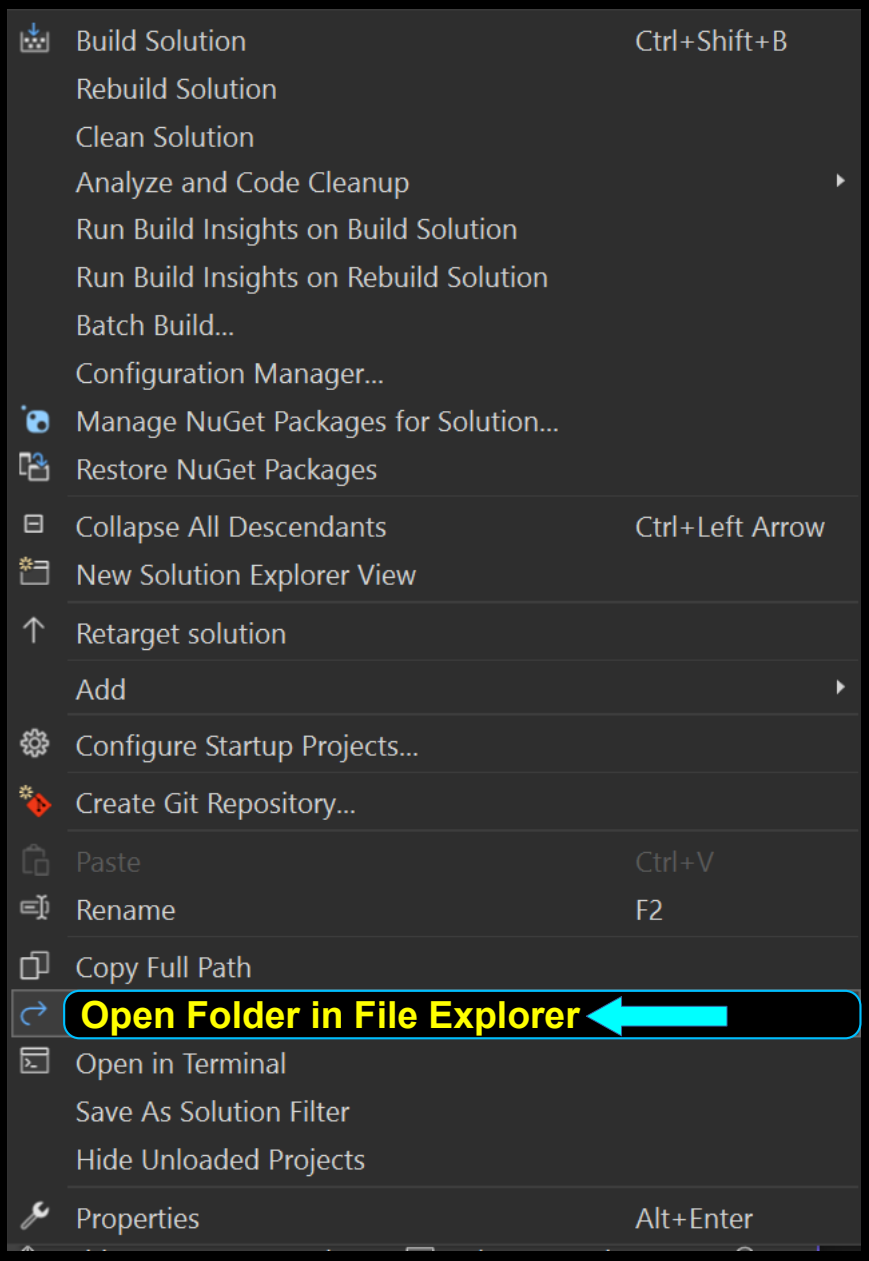
```
// Count Number of Lines in a Text File

// main.cpp

#include <iostream>
#include <fstream>
#include <string>
using namespace std;

int main()
{
    // open text file
    std::ifstream inputFile("ourTextFile.txt");

    // check if text file opened
    if (!inputFile.is_open())
    {
        std::cerr << "File won't open file."
        << "\n";

        // return an error code
        return 1;
    }
```

```
    // variable to store how many lines found in
text file
    int lineCount = 0;

    // variable to store each line of text read from
text file
    string line;

    // read text file line by line and count lines
    while (std::getline(inputFile, line))
    {
        lineCount++;
    }

    // close the file
    inputFile.close();

    // show the total number of lines
    cout << "Number of lines in the file: "
    << lineCount << "\n";

    return 0;
}
```

// How to Find Our Application .exe File

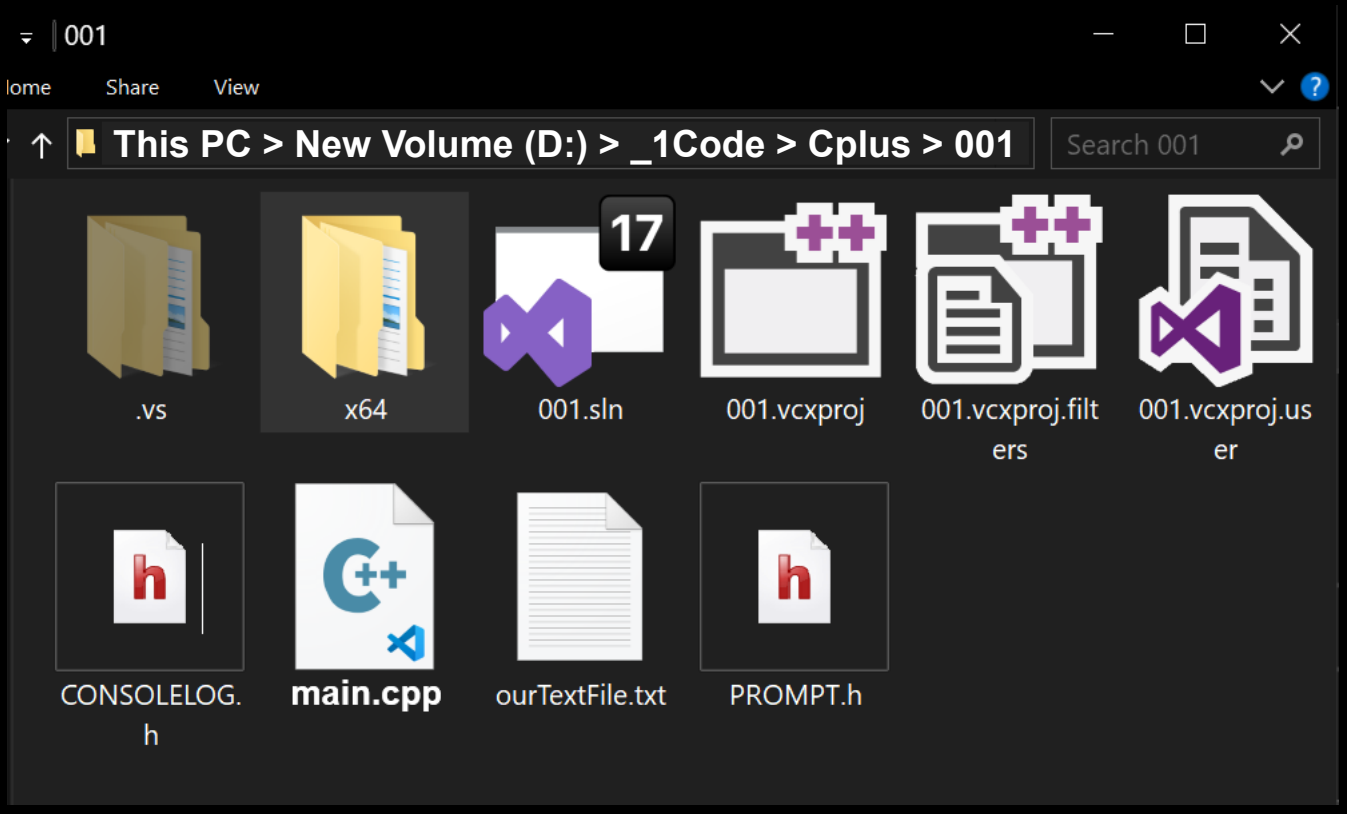
We put mouse arrow on: Solution '001' (1 of 1 project)



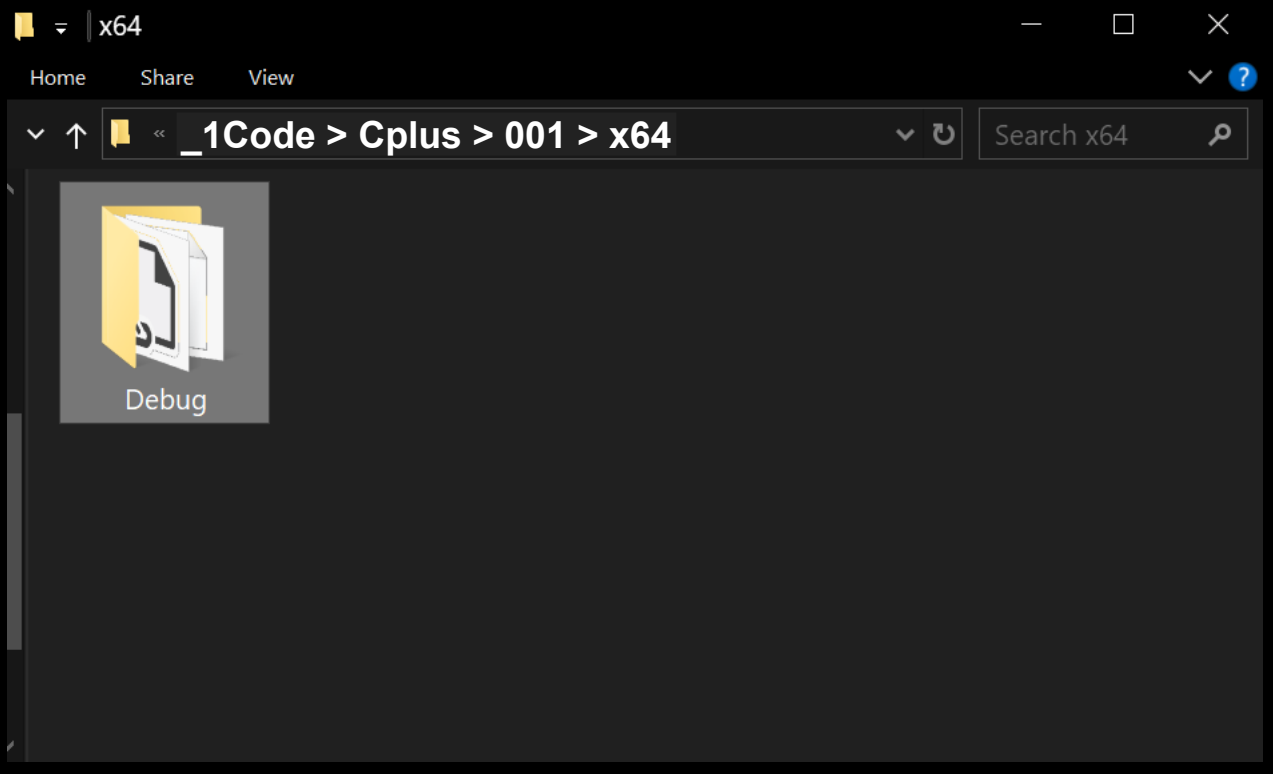


We Choose: Open Folder in File Explorer

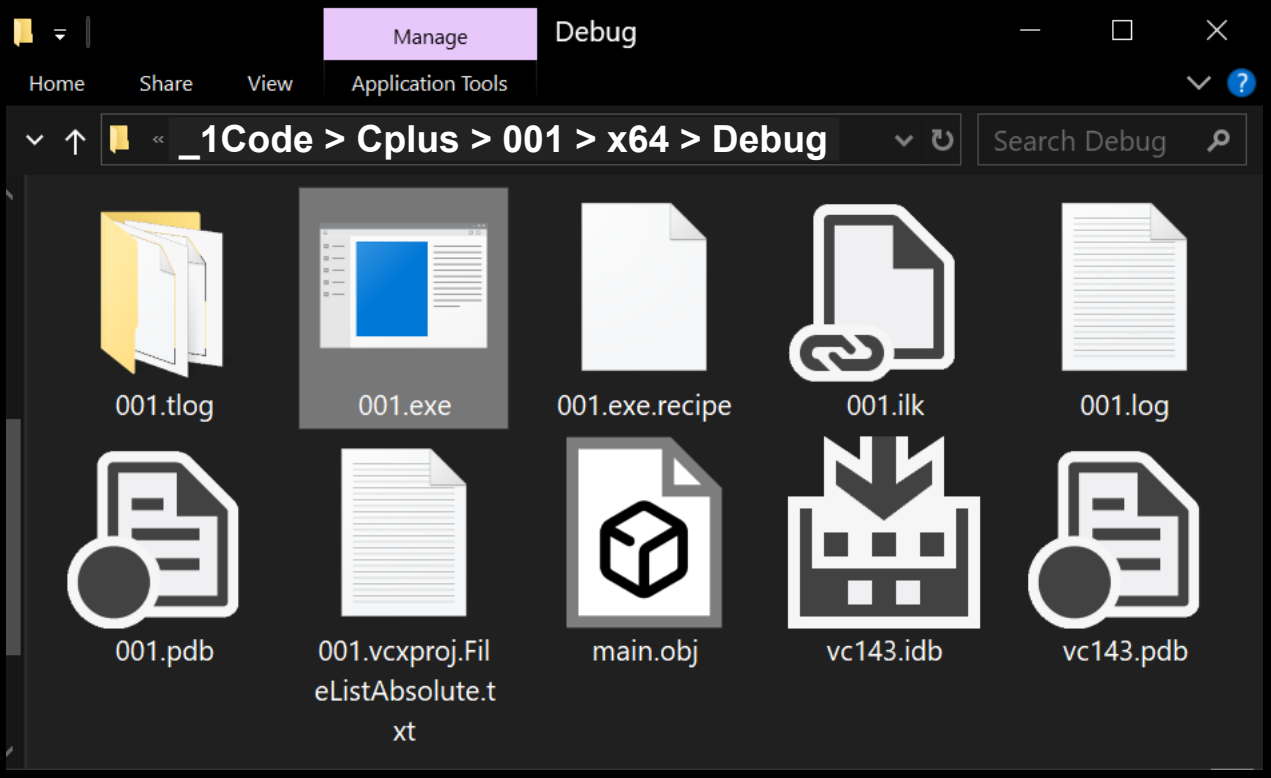
We see that our Project Folder has opened:



We Open: x64 Folder to find the Debug Folder



We Open: Debug Folder to find 001.exe



We Double Left Click: 001.exe

Our application should activate.

Happy Programming :-)

```
// Get Current Working Directory

// main.cpp

#include <iostream>
#include <windows.h>
using namespace std;

int main()
{
    wchar_t buffer[MAX_PATH];

    // get the current working directory
    if (!GetCurrentDirectoryW(MAX_PATH,
buffer))
    {
        std::wcerr << L"Error getting current
working directory" << std::endl;

        return EXIT_FAILURE;
    }

    std::wcout << L"Current working directory: "
<< buffer << std::endl;
```

```
    cout << "\nPress Enter to Exit";
    cin.get();

    return EXIT_SUCCESS;
}

/*
Current working directory: D:\_1CodeArc\
Cplus\001\x64\Debug

Press Enter to Exit
*/
```

```
// Make New Folder in Specified Location

#include <iostream>
#include <windows.h>

int main()
{
    // specify path of new folder
    LPCWSTR folderPath = L"C:\\Users\\energy\\
Desktop\\ourNewFolder";

    // create the new folder
    if (CreateDirectory(folderPath, NULL) ||
GetLastError() == ERROR_ALREADY_EXISTS)
    {
        std::wcout << L"Folder created
successfully or already exists: " << folderPath
<< std::endl;
    }
    else
    {
        std::wcerr << L"Error creating folder: "
        << folderPath << std::endl;
    }
    return 0;
}
```

```
// Make a New Folder in Documents

#include <iostream>
#include <windows.h>
#include <shlobj.h>

int main()
{
    // get path of Documents folder
    WCHAR documentsPath[MAX_PATH];
    if (SHGetFolderPathW(NULL,
CSIDL_PERSONAL, NULL, 0, documentsPath) !
= S_OK)
    {
        std::wcerr << L"Error getting Documents
path" << std::endl;
        return 1;
    }

    // append name of new folder
    std::wstring newFolderPath =
std::wstring(documentsPath) + L"\\
NewFolder7";

    // create the new folder
```

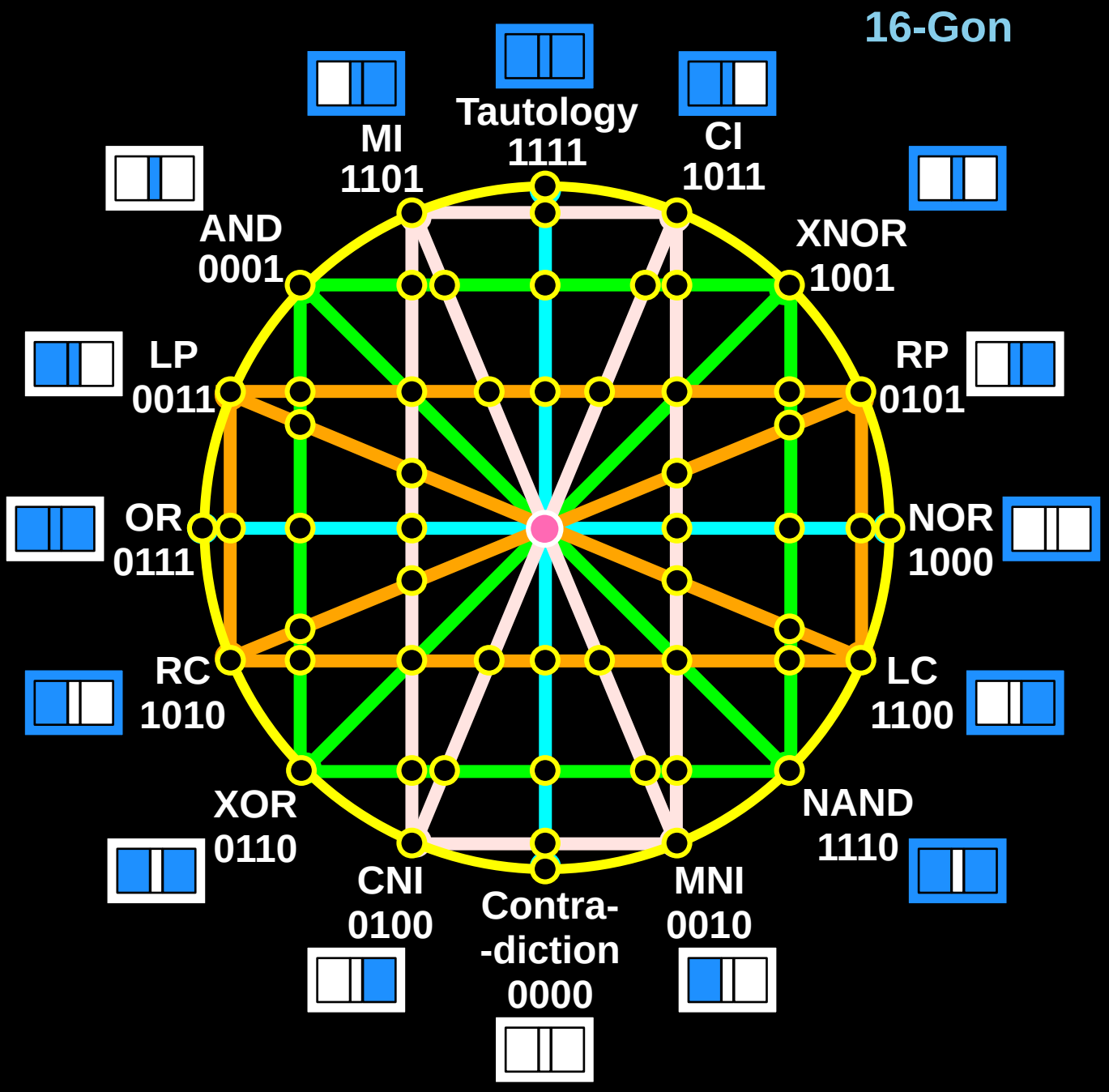
```
    if (!CreateDirectoryW(newFolderPath.c_str(),
NULL))
    {
        std::wcerr << L"Error creating folder: "
        << GetLastError() << std::endl;
        return 1;
    }

    std::wcout << L"Folder created successfully:
" << newFolderPath << std::endl;

    return 0;
}
```

```
// Make a New Folder on the Desktop

#include <iostream>
#include <windows.h>
#include <shlobj.h>

int main()
{
    // get path of Desktop folder
    WCHAR desktopPath[MAX_PATH];

    if (SHGetFolderPathW(NULL,
CSIDL_DESKTOP, NULL, 0, desktopPath) !=
S_OK)
    {
        std::wcerr << L"Error getting Desktop
path" << std::endl;
        return 1;
    }

    // append name of new folder
    std::wstring newFolderPath =
std::wstring(desktopPath) + L"\\NewFolder";

    // create the new folder
```

```
    if (!CreateDirectoryW(newFolderPath.c_str(),
NULL))
    {
        std::wcerr << L"Error creating folder: "
        << GetLastError() << std::endl;
        return 1;
    }

    std::wcout << L"Folder created successfully:
" << newFolderPath << std::endl;

    return 0;
}
```

Christopher Topalian

True Artificial Intelligence System



For More Tutorials:

CollegeOfScripting.weebly.com

CollegeOfScripting.wordpress.com

GitHub.com/ChristopherTopalian

GitHub.com/ChristopherAndrewTopalian

Youtube.com/ScriptingCollege

Twitter.com/CollegeOfScript

Rumble.com/user/CollegeOfScripting

Sites.google.com/view/CollegeOfScripting

Dedicated to God the Father

This book is created by the
College of Scripting Music & Science.

Always remember, that each time you write a script with a pencil and paper, it becomes imprinted so deeply in memory that the material and methods are learned extremely well. When you Type the scripts, the same is true.

The more you type and write out the scripts by keyboard or pencil and paper, the more you will learn programming!

Write & Type EVERY example that you find.
Keep all of your scripts organized.
Every script that you create increases your programming abilities.

SEEING CODE, is one thing,
but WRITING CODE is another.
Write it, Type it, Speak It, See It, Dream It.

www.CollegeOfScripting.weebly.com