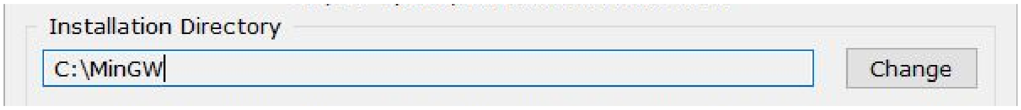
VS Code C++ Environment Setup

In this document it is assumed that have already installed VS Code. If not then go to the following link and Install VS Code first.

https://code.visualstudio.com/#alt-downloads

MinGW Compiler/Debugger

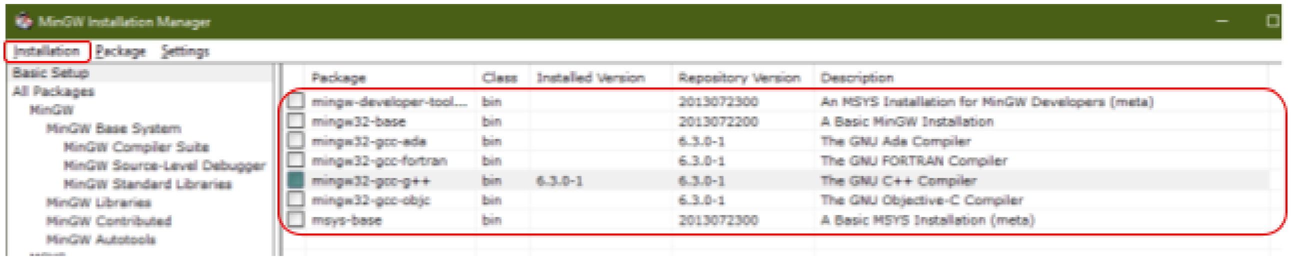
1. Go to the link https://sourceforge.net/projects/mingw/ and download MinGW compiler Setup.
2. During the initial installation make sure that installation path does **not** contain any **“wihite-spaces”.**



Installation Directory

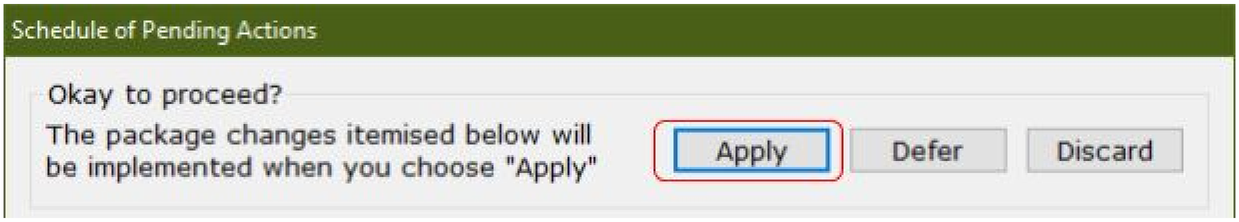
C:\MinGW

3. After the initial installation, **“MinGW Installation Manager”** will open.
    a. Select all packages from the list. (You can leave ada , forton and objc compilers unchecked).
    b. Select the **Apply Changes** from the installation menu.



	Package	Class	Installed Version	Repository Version	Description
□	mingw-developer-tool...	bin		2013072300	An MSYS Installation for MinGW Developers (meta)
□	mingw32-base	bin		2013072200	A Basic MinGW Installation
□	mingw32-gcc-ada	bin		6.3.0-1	The GNU Ada Compiler
□	mingw32-gcc-fortran	bin		6.3.0-1	The GNU FORTRAN Compiler
■	mingw32-gcc-g++	bin	6.3.0-1	6.3.0-1	The GNU C++ Compiler
□	mingw32-gcc-objc	bin		6.3.0-1	The GNU Objective-C Compiler
□	msys-base	bin		2013072300	A Basic MSYS Installation (meta)

    c. Select **Apply** from the Pop-Up window.



Schedule of Pending Actions

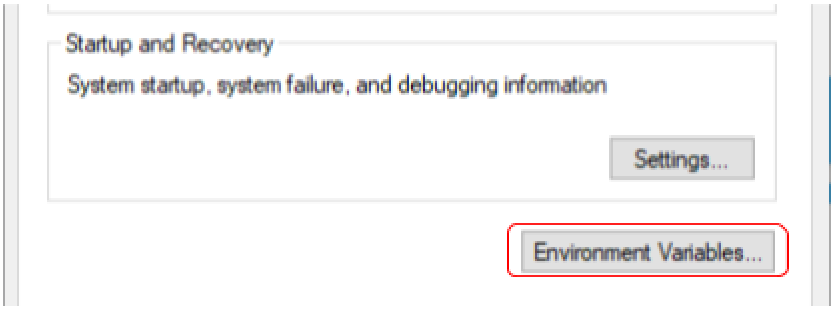
Okay to proceed?

The package changes itemised below will be implemented when you choose "Apply"

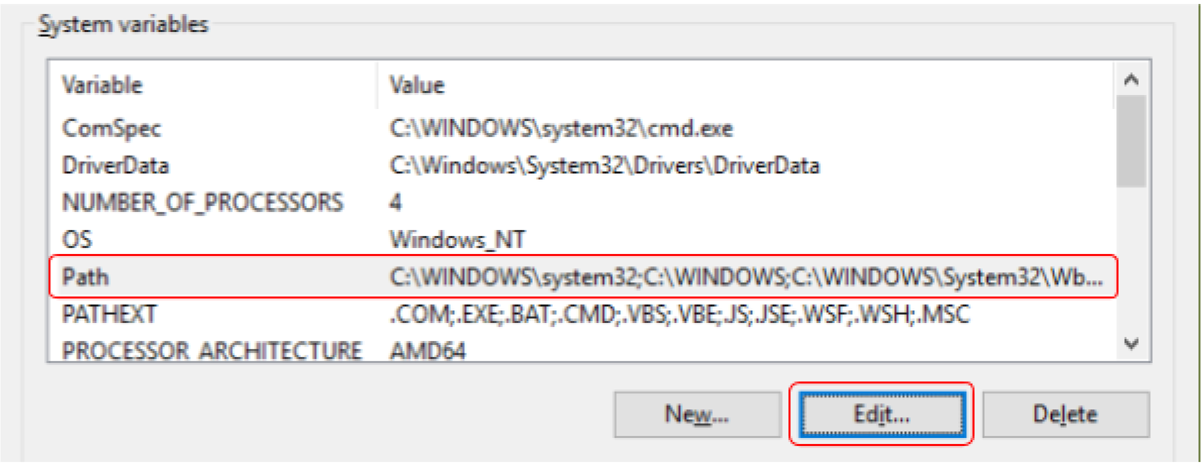
    d. Wait till everything is downloaded and installed.

Path Environment Variable

1. In the **Start Menu** type “path” and open the **“Edit the System Environment Variables”** .
2. Select Environment Variables from the opened window.

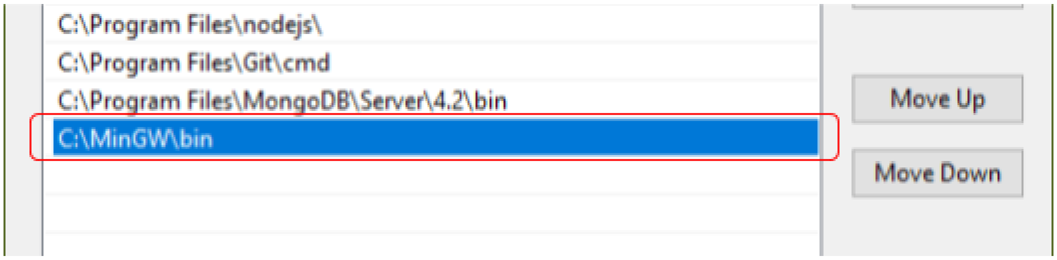


3. In the Environment Variables window find the **“path”** variable from the **System Variables** section.
   a. Select **Edit**.



Variable	Value
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wb...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64

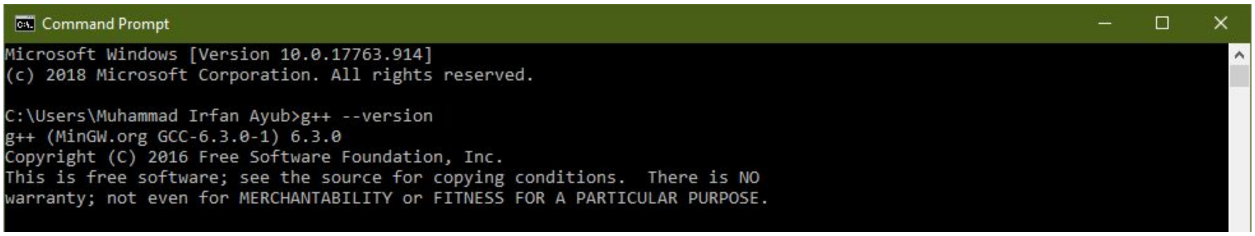
4. Add the path to the **“bin”** from the folder in the MinGW installation directory, and click OK.



C:\Program Files\nodejs\
C:\Program Files\Git\cmd
C:\Program Files\MongoDB\Server\4.2\bin
C:\MinGW\bin

NOTE: In Windows 7 put a semicolon ‘ ; ‘ and then add the path to the path string.

5. After adding the path variable check the version of the G++ compiler from command prompt to know that everything is working fine.
   a. Type **g++ --version** to check the version.



```
Microsoft Windows [Version 10.0.17763.914]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Muhammad Irfan Ayub>g++ --version
g++ (MinGW.org GCC-6.3.0-1) 6.3.0
Copyright (C) 2016 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

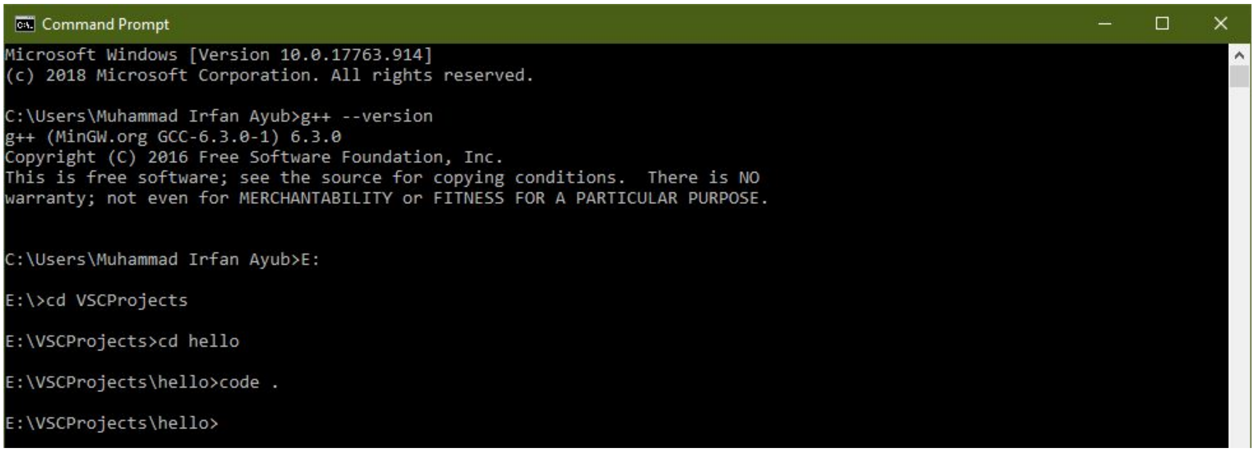
   b. Also check the debugger is correctly installed. By using **gdb --version** command.

If everything till now is working fine. Congratulations you've done a great job. :)

Setting Up VS Code

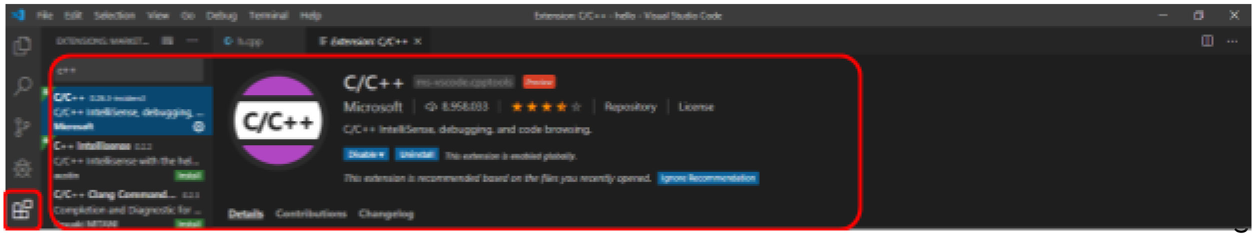
Installing the C++ Extension

1. Open VS Code in the folder in which you want to save your code.
   a. **"code ."** is the command to open the VS Code in the current folder from the command prompt.



```
Microsoft Windows [Version 10.0.17763.914]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Muhammad Irfan Ayub>g++ --version
g++ (MinGW.org GCC-6.3.0-1) 6.3.0
Copyright (C) 2016 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.


C:\Users\Muhammad Irfan Ayub>E:

E:\>cd VSCProjects

E:\VSCProjects>cd hello

E:\VSCProjects\hello>code .

E:\VSCProjects\hello>
```

2. Got to the Extensions Tab from the right.
3. Search for **C++** and install the C/C++ Extension from Microsoft.



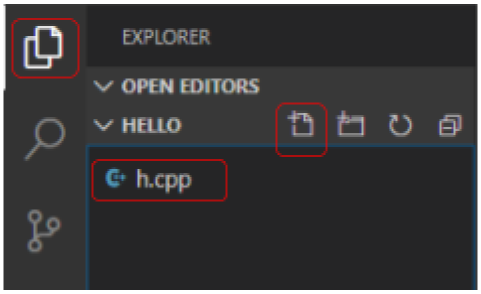
C/C++ IntelliSense, debugging, and code browsing.

This extension is enabled globally.

This extension is recommended based on the files you recently opened.

Creating the Program

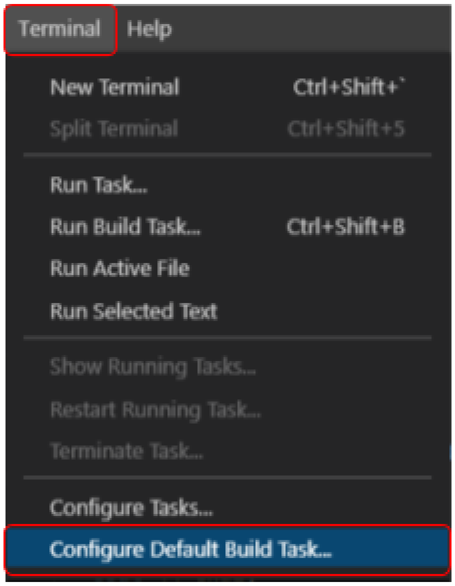
1.
2. From the **Explorer** in VS Code and create a new file named as **"h.cpp"** (you can name it whatever you want but keep the extension as **.cpp** for c++)

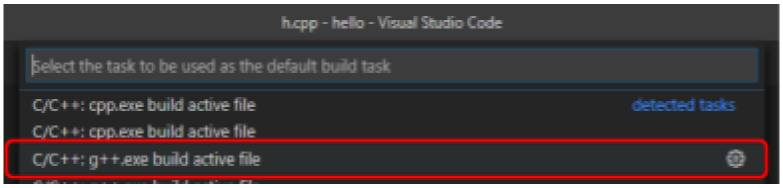


3. Write the following code in the file.

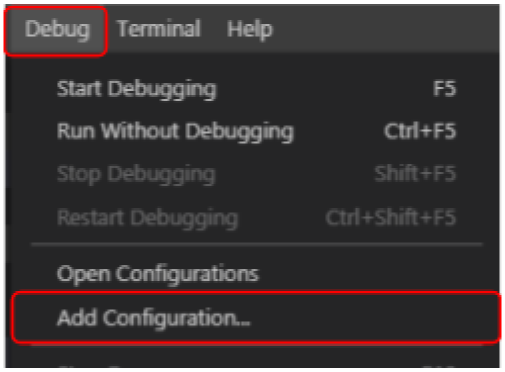
```
#include <iostream>
#include <vector>
#include <string>

using namespace std;

int main()
{
   vector<string> msg {"Hello", "C++", "World", "from", "VS Code", "and the C++ extension!"};
   for (const string& word : msg)
   {
      cout << word << " ";
   }
   cout << endl;
}
```

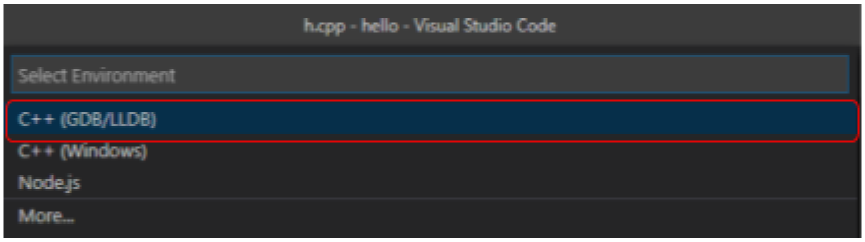
4. Now from the **Terminal** menu select **"Configure Default Build Task..."**
   a. Select **"C/C++; g++.exe build active file"** from the pop up. (Select any one if there are duplicate entries)
   b. This will create a file named **"tasks.json"**




h.cpp - hello - Visual Studio Code
Select the task to be used as the default build task
C/C++: cpp.exe build active file
detected tasks
C/C++: cpp.exe build active file
C/C++: g++.exe build active file

5. Now form the **Debug** menu select **“Add Configuration”.**
    a. Select **“C++ (GDB/LDB)”** from the pop up.
    b. This will create a file named **cpp_properties.json** or **launch.json**





Configuring the Compiler

Tasks.json:

In tasks.json make sure that the `“command”` and `“cwd”` attributes in the task array are correct and match the path of your installation.

1. In the first value of the array the `“command”` and `“cwd”` attributes have the path for `“g++.exe”.`
2. In the second value of the array the `“command”` and `“cwd”` attributes have the path for `“gcc.exe”.`

Configuring Debugger

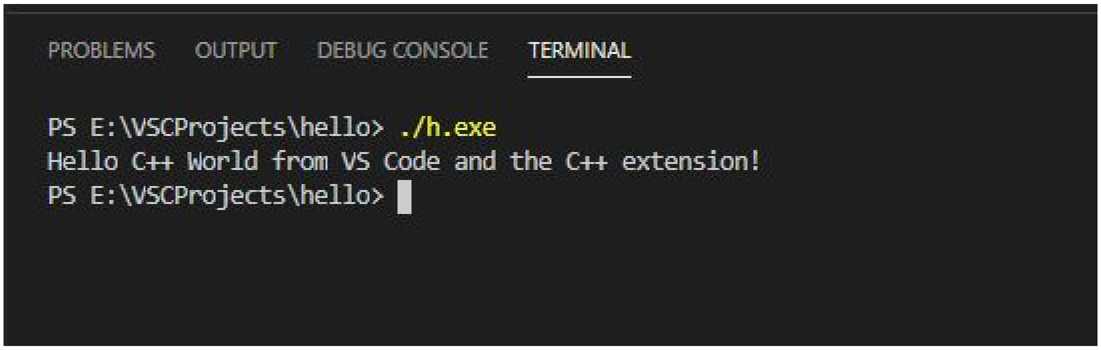
Launch.json:

In Launch.json update the `“program”` and `“miDebuggerPath”` attributes in the `‘configuration’` array as follows:

1. `"program": "${workspaceFolder}/a.exe"`
    a. This should not include any example strings.
    b. This is the path to your build (.exe) file. In this case it is named as “a.exe” by default.
    c. `${workspaceFolder}` holds the path of your current folder.
2. `"miDebuggerPath": "C:\\MinGW\\bin\\gdb",`
    a. This is the path to your debugger. (make sure that this path matches your installation of MinGW.
    b. ‘ \\ ‘ are for ignoring the escape sequences caused by single ‘ \ ‘.

Executing the Program

1. To execute the program you can press `Ctrl + Shift + B` (it is for the command “Run Build Task”). You can also find it in Terminal Menu.
2. After the program has been built with no errors you will now see a new file named as **‘h.exe’** is created in the Explorer tab. This is the output file for you program.
3. To run the exe file in the Terminal of VS Code run `“./h.exe” command.` Now the terminal will show the output of the code.



```
PS E:\VSCProjects\hello> ./h.exe
Hello C++ World from VS Code and the C++ extension!
PS E:\VSCProjects\hello>
```

4. Make sure that the terminal is in the same directory as your .cpp file.

For More Debugging Details Refer to the Following Link.

https://code.visualstudio.com/docs/cpp/config-mingw

NOTE: This link installs mingw compiler without using MinGW package manager. It also works fine but it only changes the installation path for mingw.

Thanks for Reading. Happy Coding :)