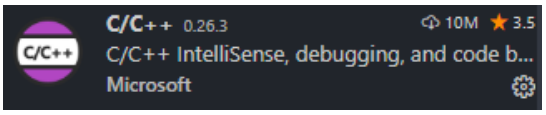
# Installing C++ for VSCode:

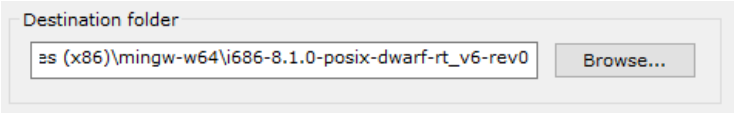
More info: https://code.visualstudio.com/docs/cpp/config-mingw

Steps:

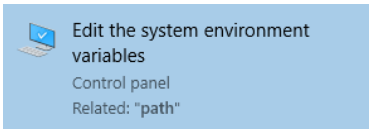
1. Download the C++ VSCode extension



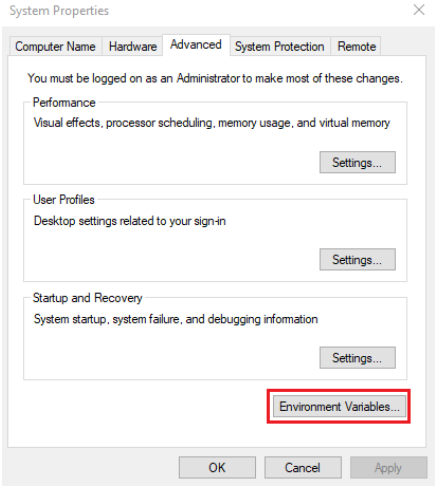

2. Install MinGW:
   a. https://sourceforge.net/projects/mingw-w64/files/Toolchains%20targetting%20Win32/Personal%20Builds/mingw-builds/installer/mingw-w64-install.exe/download
   b. Take note of your installation path (needs to be added to path)



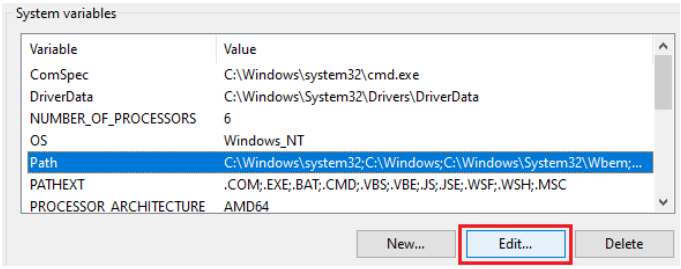

3. Add the "bin" directory of your MinGW installation to path
   a. Search path in windows



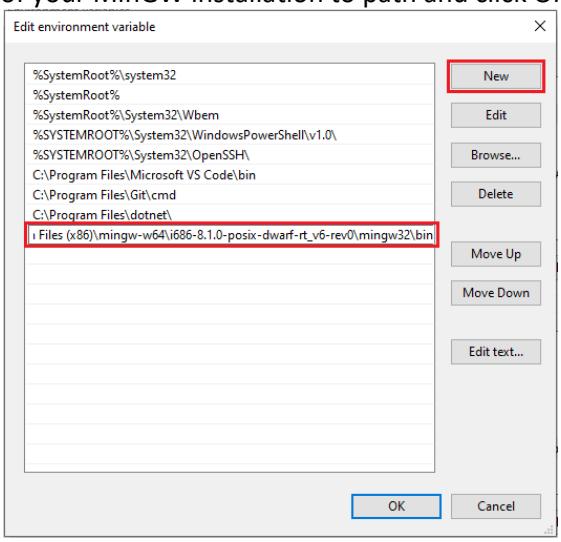

   b. In the "Advanced" tab, click the "Environment Variables" button




   c. Under "System variables" edit "Path"



d. Add the “bin” folder of your MinGW installation to path and click OK




4. Check your MigGW installation
   a. Run `g++ --version` in command prompt

```
g++ (i686-posix-dwarf-rev0, Built by MinGW-W64 project) 8.1.0
Copyright (C) 2018 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.
```

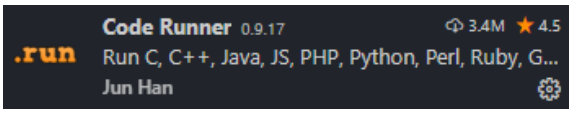
   b. Run `gdb --version` in command prompt

```
GNU gdb (GDB) 8.1
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "i686-w64-mingw32".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word".
```

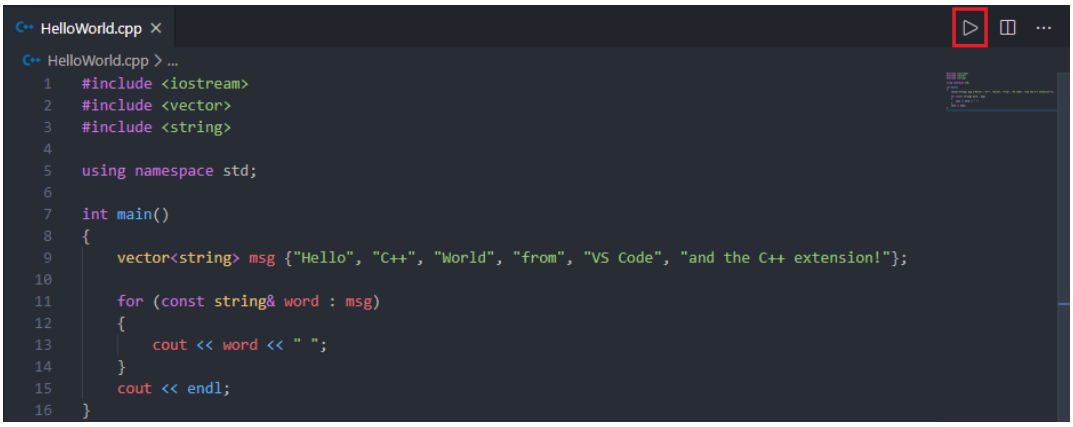
5. If already open, restart VSCode. Open a folder in VSCode and make a HelloWorld.cpp script using the following: (A copy of this script can be found in the “more info” link at the top)

```
#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;
}
```

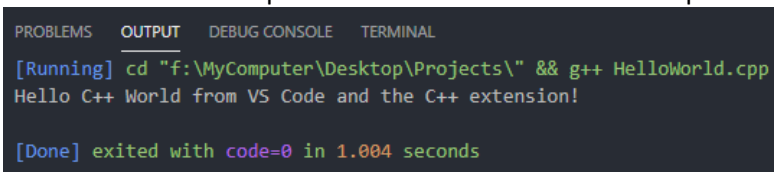
6. Download the Code Runner extension



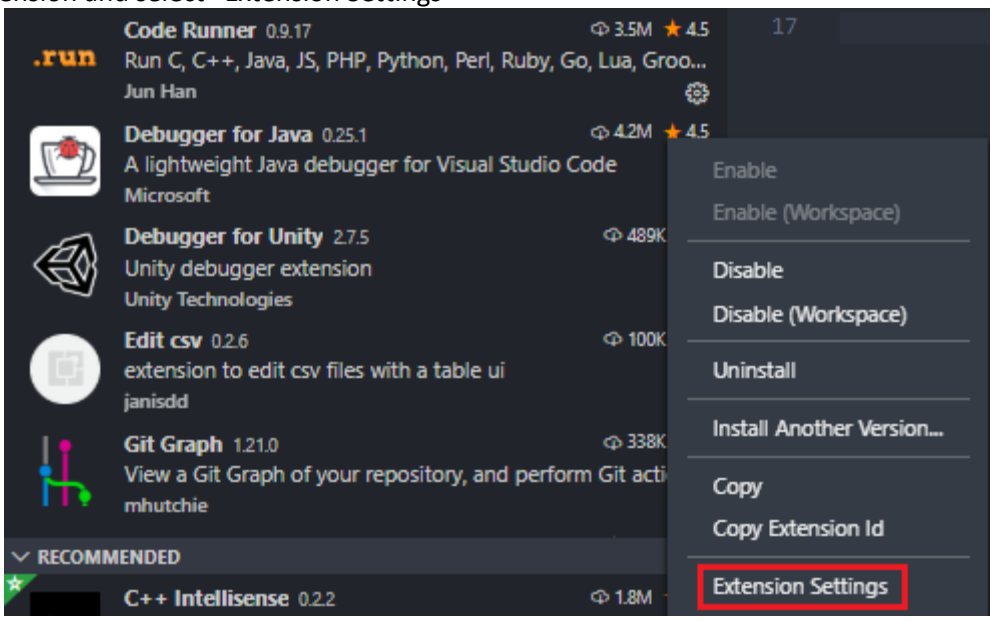

7. Click the Code Runner run button



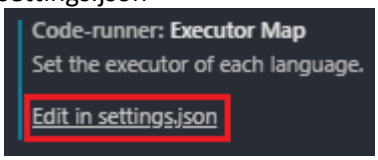

8. Celebrate the fact that Code Runner has optimized the tedious C++ code compilation and running process




9. However, a few more steps are required to run more complex OOP code. Click the gear symbol on the code runner extension and select “Extension Settings”

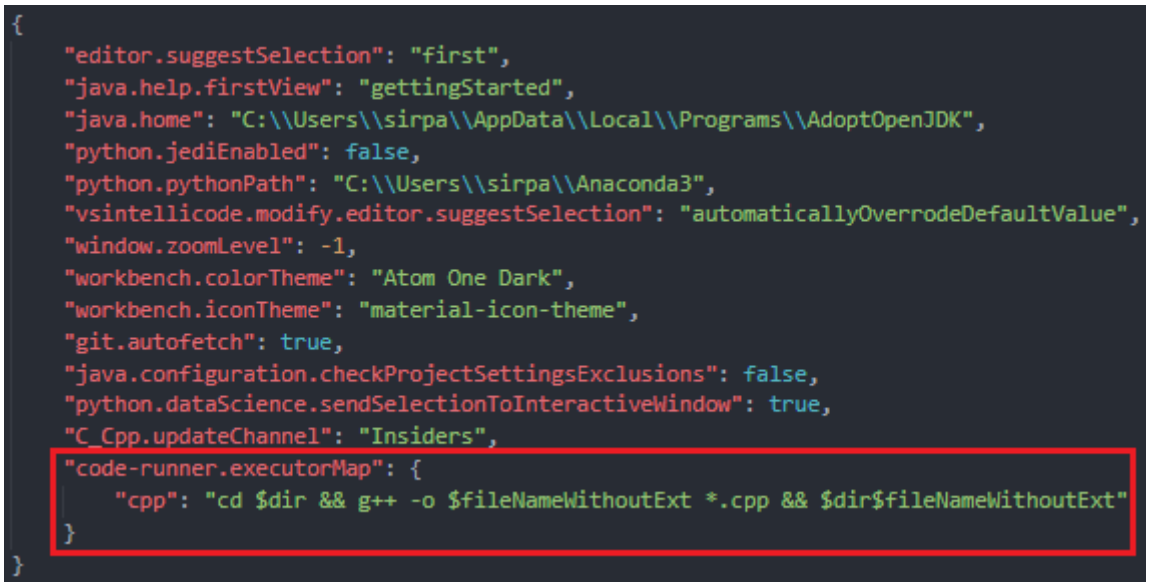



10. Under Executor Map, select “Edit in settings.json”



11. Copy and paste the following line into the JSON file as shown and save:

```
"code-runner.executorMap": {
        "cpp": "cd $dir && g++ -o $fileNameWithoutExt *.cpp && $dir$fileNameWithoutExt"
    }
```




12. Hope that C++ is now fully functional and that there will be no more future additions to this tutorial