



# Programming in C++

ICS 45C



Caio Batista de Melo

# Install things



Install VS Code: <https://code.visualstudio.com/docs/setup/setup-overview>

Install the Microsoft C++ extension:

<https://marketplace.visualstudio.com/items?itemName=ms-vscode.cpptools>

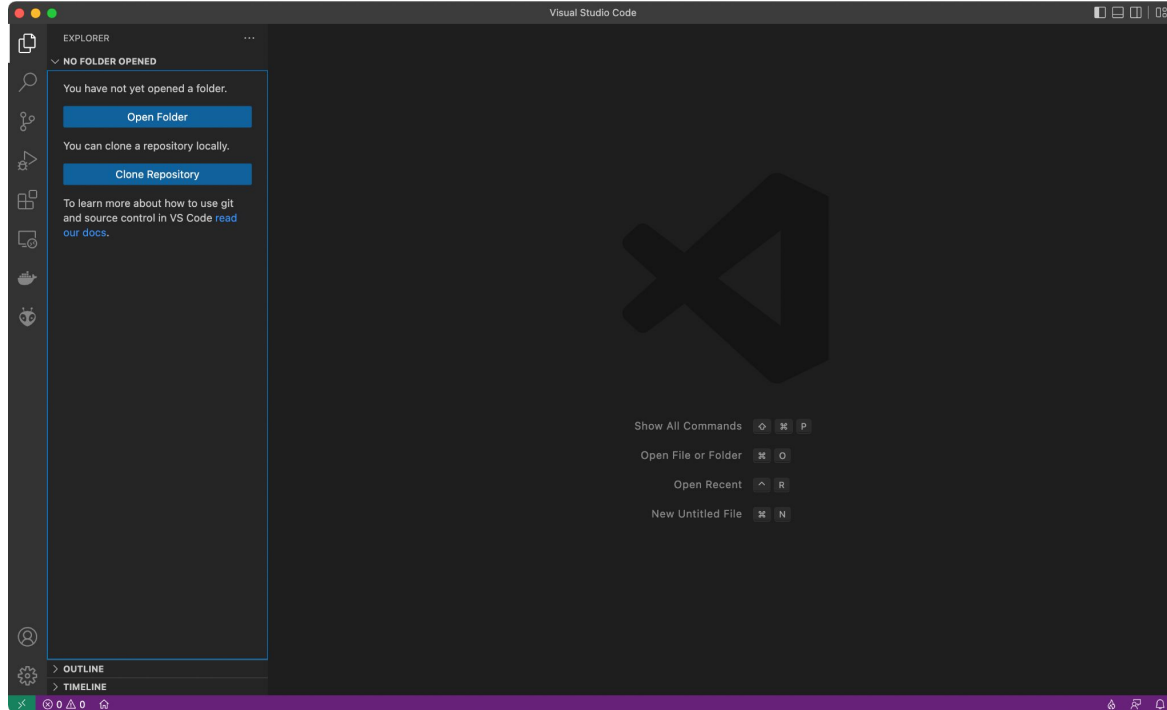
[optional] Install a separate clang-format: <https://pypi.org/project/clang-format/>



# Setup code formatting



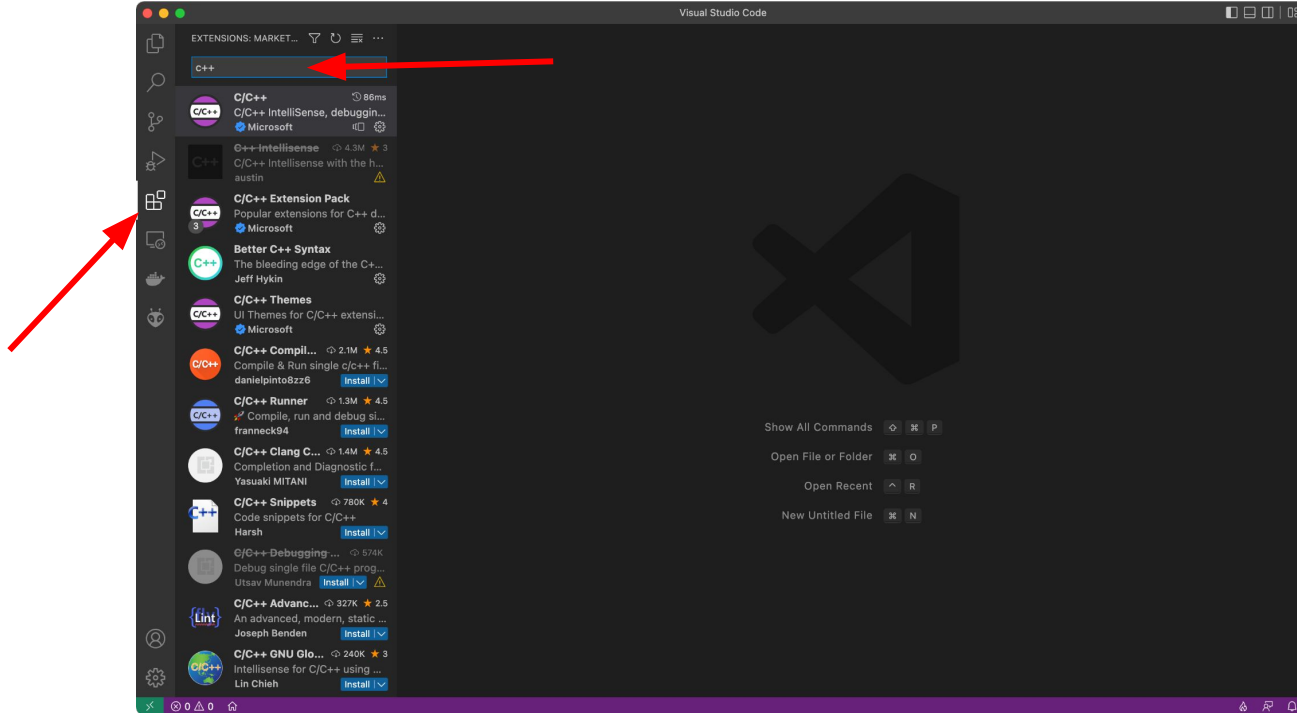
First, open VS Code:



# Setup code formatting



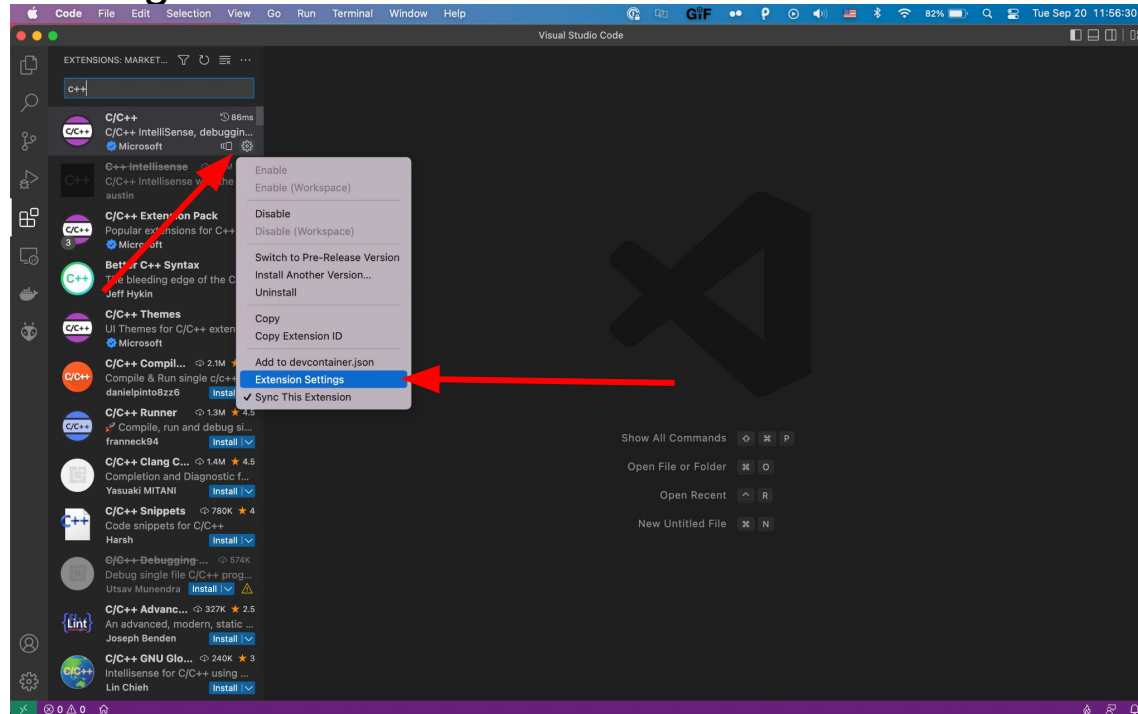
Find the extension:



# Setup code formatting



Go to extension settings:



# Setup code formatting



Search “clang-format” settings:

The screenshot shows the VS Code Settings window with the search bar containing "@ext:ms-vscode.cpptools clang-format". The search results show 4 settings found. A red arrow points to the search bar. The settings are:

- User**
- Extensions (4)**
  - C/C++**
  - C/C++ Code Analysis > Clang Tidy > Code Action: Format Fixes**
    - If 'true', formatting will be run on the lines changed by 'Fix' code actions.
  - C/C++ Clang\_format\_fallbackStyle**

Name of the predefined style used as a fallback in case clang-format is invoked with style file but the .clang-format file is not found. Possible values are Visual\_Studio, LLVM, Google, Chromium, Mozilla, WebKit, Microsoft, GNU, none, or use {key: value, ...} to set specific parameters. For example, the Visual\_Studio style is similar to: { BasedOnStyle: LLVM, UseTab: Never, IndentWidth: 4, TabWidth: 4, BreakBeforeBraces: Allman, AllowShortIfStatementsOnASingleLine: false, IndentCaseLabels: false, ColumnLimit: 0, AccessModifierOffset: -4, NamespaceIndentation: All, FixNamespaceComments: false }.

Visual Studio
  - C/C++ Clang\_format\_path (Not synced)**

The full path of the clang-format executable. If not specified, and clang-format is available in the environment path, that is used. If not found in the environment path, the clang-format bundled with the extension will be used.
  - C/C++ Clang\_format\_style**

Coding style, currently supports: Visual\_Studio, LLVM, Google, Chromium, Mozilla, WebKit, Microsoft, GNU. Use file to load the style from a .clang-format file in the current or parent directory. Use {key: value, ...} to set specific parameters. For example, the Visual\_Studio style is similar to: { BasedOnStyle: LLVM, UseTab: Never, IndentWidth: 4, TabWidth: 4, BreakBeforeBraces: Allman, AllowShortIfStatementsOnASingleLine: false, IndentCaseLabels: false, ColumnLimit: 0, AccessModifierOffset: -4, NamespaceIndentation: All, FixNamespaceComments: false }.

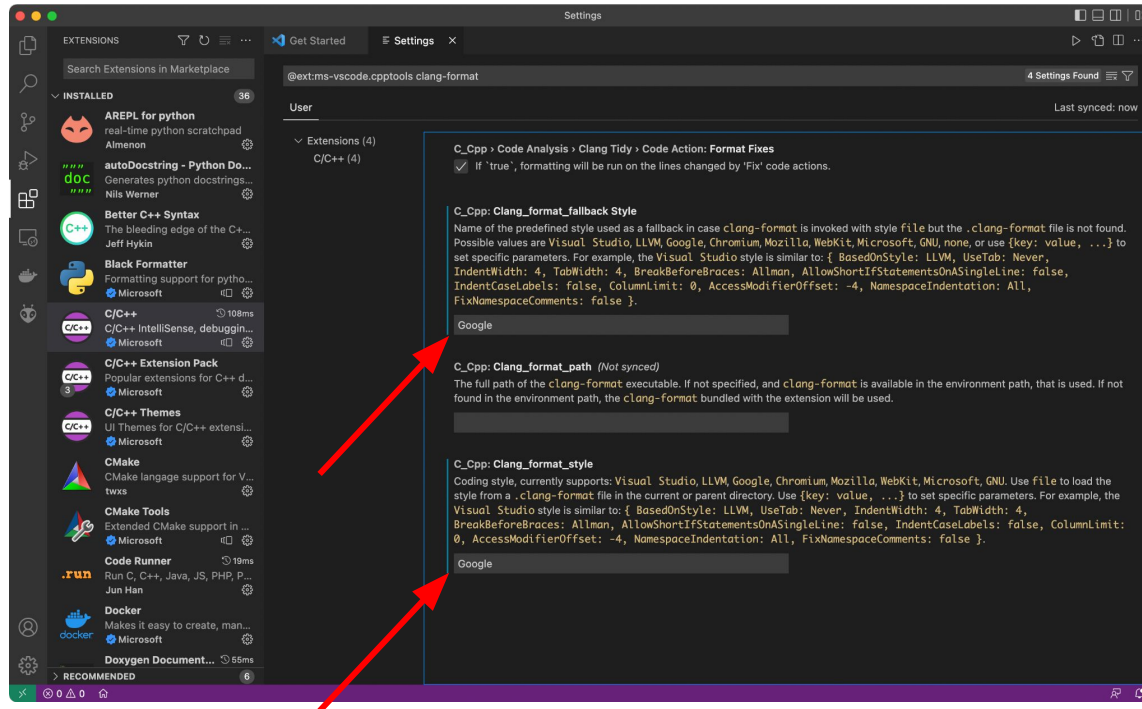
Google



# Setup code formatting



Change the format style to “Google” and save (cmd + s, ctrl + s):



# Formatting your code



Open your file:

```
example.cpp  
Get Started example.cpp x  
Users > caio > Desktop > example.cpp  
1 #include <iostream>  
2     using namespace std;  
3  
4     int main(){  
5         cout << "Hello world!" << endl;  
6         return 0;}  
7
```





# Formatting your code



Open command palette (cmd/ctrl + shift + p) and use “>format document”:

A screenshot of the Visual Studio Code editor interface. The title bar shows "example.cpp". The command palette is open, displaying the search input ">format document" and a list of results. The top result is "Format Document", which is highlighted. Below it, there is a sub-menu titled "Format Document With..." with a search input and a list of options. The code editor in the background shows a C++ program with the following content:

```
1 #include <string>
2 using namespace std;
3
4 int main(){
5     cout << "Hello world!" << endl;
6     return 0;}
7
```

The status bar at the bottom indicates "Ln 7, Col 1 Spaces: 4 UTF-8 LF C++ Mac".

# Formatting your code



Your file should be nicely formatted:

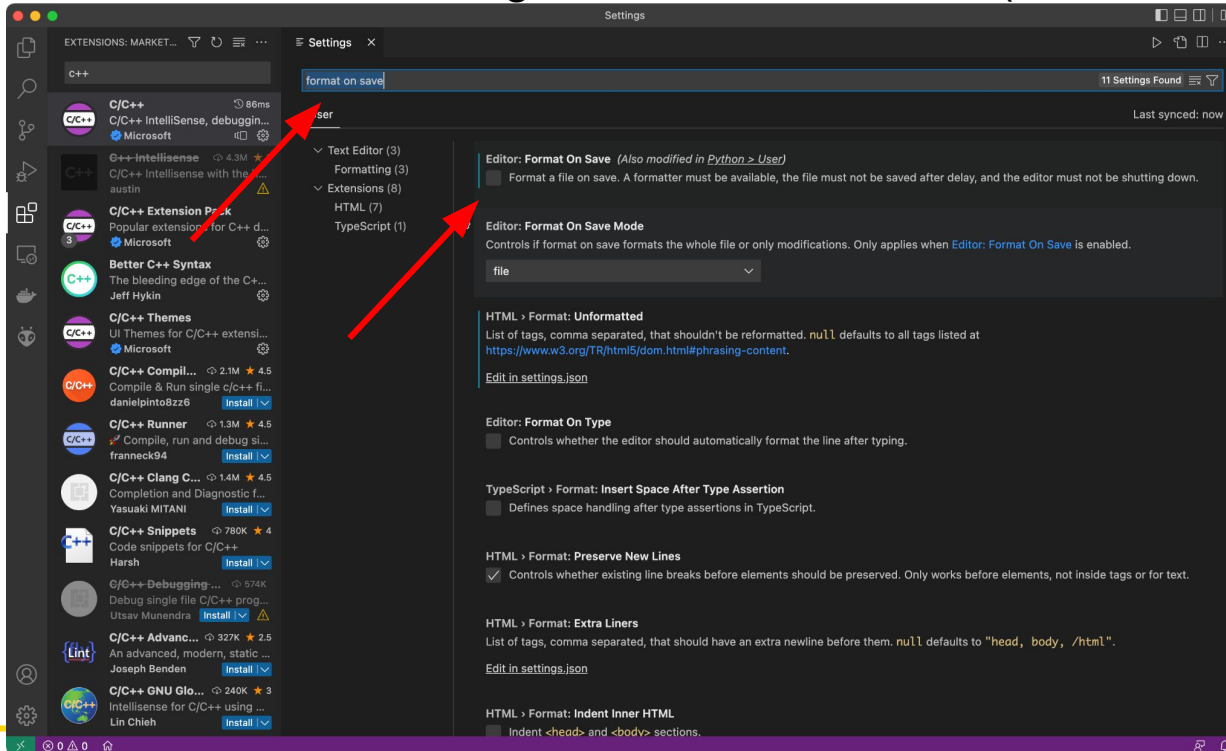
A screenshot of a code editor window titled "example.cpp". The editor shows a C++ program with the following code:

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      cout << "Hello world!" << endl;
6      return 0;
7  }
8
```

The code is formatted with proper indentation and line numbers. The editor interface includes a sidebar with icons for file operations, search, and settings, and a status bar at the bottom showing "Ln 8, Col 1", "Spaces: 2", "UTF-8", "LF", "C++", "Mac", and a notification bell.

# [OPTIONAL] Setup auto formatting when saving

Search for “format on save” on settings, enable it, and save (cmd + s, ctrl + s):



Let us know if you have any questions!