

---

Department of Electrical and Computer Engineering  
Spring 2022

# Accelerated Object Oriented Programming (CS 1420)

## 1.1 Programming Environment Setup

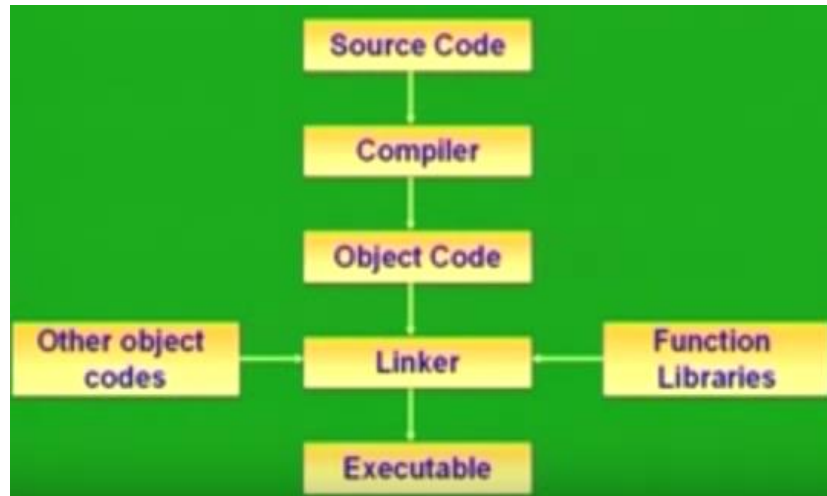
---

Habtamu Minassie  
Habtamu.aycheh@Utah.edu

# Getting Started

## ■ Recap

### □ Creating and Running Programs



## ■ Before starting writing programming ( E.g. C++), the requirements are:

- Source code editor and
- C++ compiler

✓ IDE

# Integrated Development Environment (IDE)

- IDE is a programming environment that has **integrated** into software application and provides a GUI builder such that
  - **Code editor**
  - **Compiler, Linker and Loader**
- Examples



<https://docs.microsoft.com/en-us/visualstudio/?view=vs-2022>

Visual Studio



Code:: Blocks

<http://www.codeblocks.org/>



Eclipse CDT

<https://www.eclipse.org/cdt/>



Xcode

<https://developer.apple.com/xcode/>



Visual Studio Code

<https://code.visualstudio.com/>



Dev C++

<https://www.bloodshed.net/>

# There are also online C++ Editors

[https://www.onlinegdb.com/online\\_cplusplus\\_compiler](https://www.onlinegdb.com/online_cplusplus_compiler)



The screenshot displays the Online C++ Compiler interface. The top section is a code editor with a dark background, showing a C++ program in a file named `main.cpp`. The code includes a multi-line comment explaining the compiler's purpose and a simple `main` function that prints "Hello World". The bottom section is a terminal window with a dark background, showing the output of the program: "Hello World" and a confirmation message that the program finished with exit code 0. A green circular icon with a white speech bubble is visible in the bottom right corner of the terminal window.

```
1  /*****  
2  
3      Online C++ Compiler.  
4      Code, Compile, Run and Debug C++ program online.  
5      Write your code in this editor and press "Run" button to compile and execute it.  
6  
7      *****/  
8  
9  #include <iostream>  
10  
11  int main()  
12  {  
13      std::cout<<"Hello World";  
14  
15      return 0;  
16  }  
17
```

input

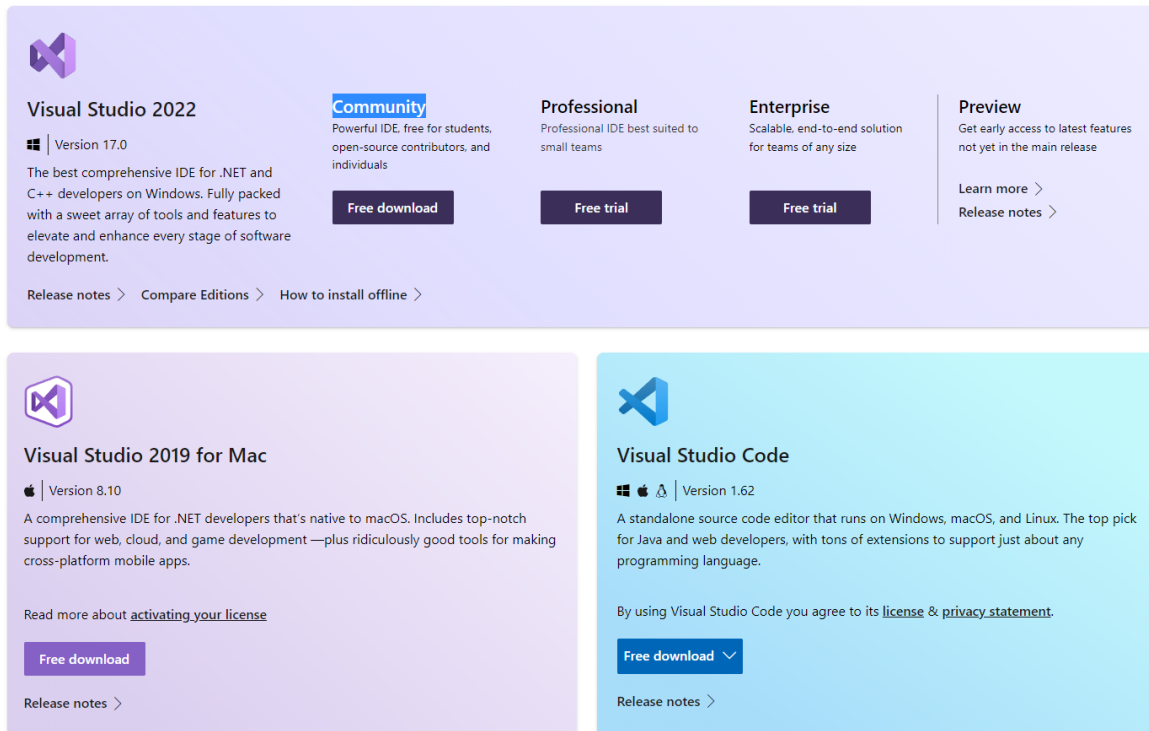
Hello World

...Program finished with exit code 0  
Press ENTER to exit console.

# Install Visual Studio

## ■ Step 1: Download the Software (Community Edition)

❑ <https://docs.microsoft.com/en-us/visualstudio/?view=vs-2022>



The screenshot displays the Visual Studio download page. At the top, the Visual Studio logo is shown. Below it, the main heading is "Visual Studio 2022" with a subheading "Version 17.0". The description states: "The best comprehensive IDE for .NET and C++ developers on Windows. Fully packed with a sweet array of tools and features to elevate and enhance every stage of software development." Below this, there are links for "Release notes", "Compare Editions", and "How to install offline".

Four editions are listed in a row:

- Community**: Powerful IDE, free for students, open-source contributors, and individuals. A "Free download" button is present.
- Professional**: Professional IDE best suited to small teams. A "Free trial" button is present.
- Enterprise**: Scalable, end-to-end solution for teams of any size. A "Free trial" button is present.
- Preview**: Get early access to latest features not yet in the main release. Links for "Learn more" and "Release notes" are present.

Below the main section, there are two additional product cards:

- Visual Studio 2019 for Mac**: Version 8.10. A comprehensive IDE for .NET developers that's native to macOS. Includes top-notch support for web, cloud, and game development—plus ridiculously good tools for making cross-platform mobile apps. A "Free download" button is present. Links for "Release notes" and "Read more about activating your license" are also shown.
- Visual Studio Code**: Version 1.62. A standalone source code editor that runs on Windows, macOS, and Linux. The top pick for Java and web developers, with tons of extensions to support just about any programming language. A "Free download" button is present. Links for "Release notes" and "By using Visual Studio Code you agree to its license & privacy statement" are also shown.

# Choose workloads

Modifying — Visual Studio Community 2022 — 17.0.5

Workloads Individual components Language packs Installation locations

## Web & Cloud (4)



### ASP.NET and web development

Build web applications using ASP.NET Core, ASP.NET, HTML/JavaScript, and Containers including Docker supp...



### Azure development

Azure SDKs, tools, and projects for developing cloud apps and creating resources using .NET and .NET Framework...



### Python development

Editing, debugging, interactive development and source control for Python.



### Node.js development

Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.



## Desktop & Mobile (5)



### Mobile development with .NET

Build cross-platform applications for iOS, Android or Windows using Xamarin.



### .NET desktop development

Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET and .NET Frame...



### Desktop development with C++

Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild.



### Universal Windows Platform development

Create applications for the Universal Windows Platform with C#, VB, or optionally C++.



## Installation details

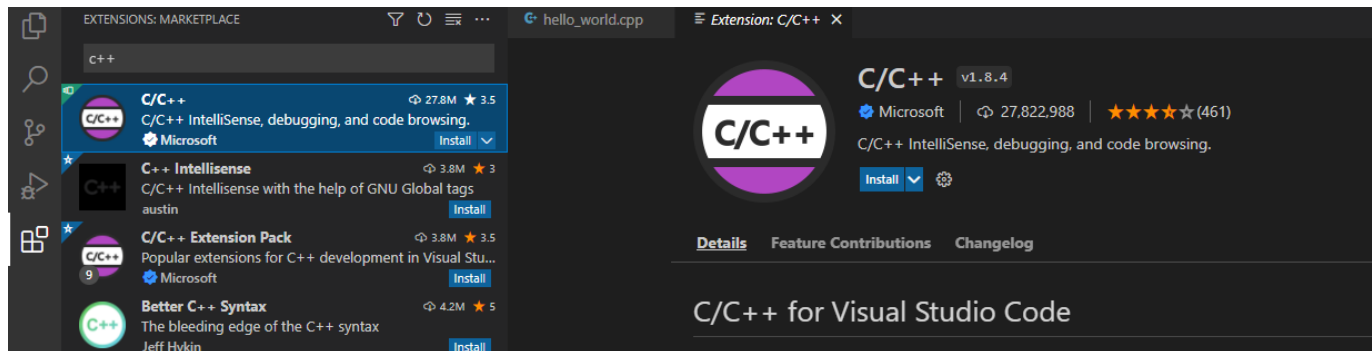
### ✓ C++ core desktop features

#### ▼ Optional

- ☒ MSVC v143 - VS 2022 C++ x64/x86 build t...
- ☒ Windows 10 SDK (10.0.19041.0)
- ☒ Just-In-Time debugger
- ☒ C++ profiling tools
- ☒ C++ CMake tools for Windows
- ☒ C++ ATL for latest v143 build tools (x86 &...
- ☒ Test Adapter for Boost.Test
- ☒ Test Adapter for Google Test
- ☒ Live Share
- ☒ IntelliCode
- ☒ C++ AddressSanitizer
- ☐ MSVC v143 - VS 2022 C++ ARM64 build t...
- ☐ C++ MFC for latest v143 build tools (x86...
- ☐ C++/CLI support for v143 build tools (Late...
- ☐ C++ Modules for v143 build tools (x64/x8...
- ☒ C++ Clang tools for Windows (12.0.0 - x64...
- ☐ JavaScript diagnostics
- ☐ Incredibuild - Build Acceleration
- ☐ Windows 11 SDK (10.0.22000.0)

# Install Visual Studio Code

- <https://code.visualstudio.com/Download>
- Install the C/C++ extension for VS Code
  - install the C/C++ extension by searching for 'c++' in the Extensions view (Ctrl+Shift+X).
    1. Open VS Code.
    2. Select the Extensions view icon on the Activity bar or use the keyboard shortcut (Ctrl+Shift+X).
    3. Search for 'C++'.
    4. Select Install.



# Install the C++ compiler

## ■ Step 1: Download MinGW

- MinGW is a compiler system based on the GNU GCC and Binutils projects that compiles and links code to be run on Win32 (Windows) systems. It provides C, C++ and Fortran compilers plus other related tools. 'MinGW' refers to the "Minimalist GNU for Windows" project
- <https://nuwen.net/mingw.html>

### Download

My MinGW distribution ("distro") is **x64-native** and currently contains **GCC 11.2.0** and **Boost 1.77.0**.

[mingw-18.0.exe](#) (96.9 MB) : This is a self-extracting archive. It's incredibly easy to install; see [How To Install](#) below.

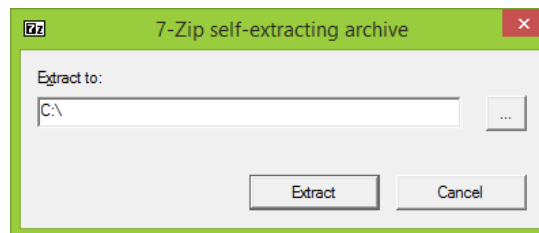


[mingw-18.0-without-git.exe](#) (49.3 MB) : This is smaller, if you've already installed git.

My build scripts are [available on GitHub](#), and they're also stored within the distro itself.

## ■ Step 2: How to Install

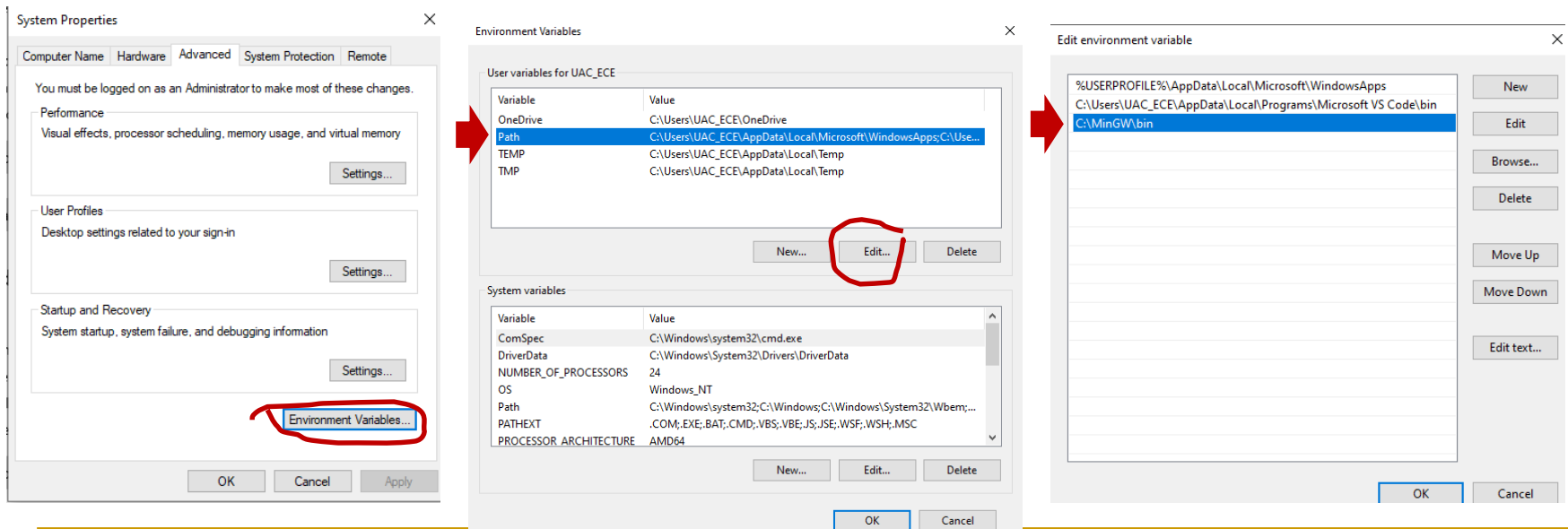
- Run the self-extracting archive, for instance it'll create C:\MinGW





# Set Environmental path

- Add the path to your Mingw bin folder to the Windows PATH environment variable by using the following steps:
  1. In the Windows search bar, type 'settings' to open your Windows Settings.
  2. Search for Edit environment variables for your account.
  3. Choose the Path variable in your User variables and then select Edit.
  4. Select New and add the Mingw destination folder path to the system path. The exact path depends on where you installed it.
    - Add this to the path: C:\mingw\bin.
  5. Select OK to save the updated PATH. You will need to reopen any console windows for the new PATH location to be available



# Check your MinGW installation

- Open a new Command Prompt and type:

```
g++ --version  
gdb --version
```

- If you don't see the expected output or g++ or gdb is not a recognized command, make sure your PATH entry matches the MinGW binary location

```
C:\Users\UAC_ECE>g++ --version  
g++ (GCC) 11.2.0  
Copyright (C) 2021 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\UAC_ECE>g++ --version  
g++ (GCC) 11.2.0  
Copyright (C) 2021 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.
```

# Utah CADE Machines

- All of you should have access to CADE servers
  - If not, go apply for one at: <https://www.cade.utah.edu/>
- CADE machines have all the facilities we need
  - Domain: labX-Y.eng.utah.edu
  - X = 1, 2, 3, ... (# of lab space)
  - Y = 1, 2, 3, 4, 5, ... (machine # of each lab space)
  - For example, lab2-20.eng.utah.edu
  - Account: **your uID**
  - Password: **your uID login password**
- Remote login using ssh (the easiest way)
  - `ssh -x u6024634@lab2-20.eng.utah.edu`

# Compilation and Linking Demo: Lab 1

## ■ Use g++ or clang compiler :

```
Developer Command Prompt for VS 2022

Directory of D:\CPP
02/04/2022  04:59 PM  <DIR>      .
02/04/2022  04:59 PM  <DIR>      ..
02/04/2022  12:07 PM                351 hello_world.cpp
               1 File(s)                351 bytes
               2 Dir(s)  155,738,972,160 bytes free

D:\CPP>clang -c -o hello_world.obj hello_world.cpp

D:\CPP>clang -o hello_world.exe hello_world.obj

D:\CPP>hello_world.exe
Hello C++

D:\CPP>dir
Volume in drive D is Data
Volume Serial Number is A8AC-BCB9

Directory of D:\CPP
02/04/2022  05:01 PM  <DIR>      .
02/04/2022  05:01 PM  <DIR>      ..
02/04/2022  12:07 PM                351 hello_world.cpp
02/04/2022  05:01 PM          181,760 hello_world.exe
02/04/2022  05:00 PM          62,314 hello_world.obj
               3 File(s)            244,425 bytes
               2 Dir(s)  155,738,722,304 bytes free

D:\CPP>
```

g++ Linux man page

<https://linux.die.net/man/1/g++>

```
Command Prompt

2 Dir(s)  155,738,722,304 bytes free

D:\CPP>g++ --version
g++ (GCC) 11.2.0
Copyright (C) 2021 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.

D:\CPP>g++ -c -o hello_world.obj hello_world.cpp

D:\CPP>g++ -o hello_world.exe hello_world.obj

D:\CPP>hello_world
Hello C++

D:\CPP>dir
Volume in drive D is Data
Volume Serial Number is A8AC-BCB9

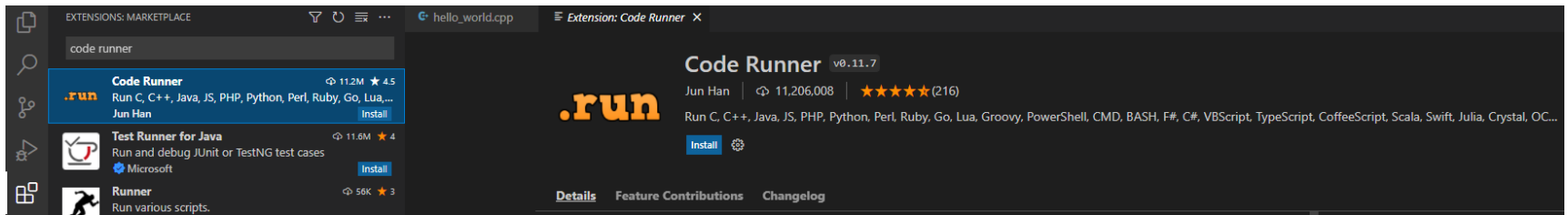
Directory of D:\CPP
02/04/2022  05:10 PM  <DIR>      .
02/04/2022  05:10 PM  <DIR>      ..
02/04/2022  12:07 PM                351 hello_world.cpp
02/04/2022  05:10 PM          2,965,139 hello_world.exe
02/04/2022  05:10 PM          1,839 hello_world.obj
               3 File(s)            2,967,329 bytes
               2 Dir(s)  155,735,752,704 bytes free
```

```
Windows PowerShell

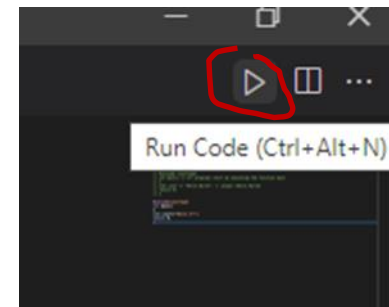
PS D:\CPP>
PS D:\CPP> g++ -c -o hello_world.obj hello_world.cpp
PS D:\CPP> g++ -o hello_world hello_world.obj
PS D:\CPP> ./hello_world
Hello World
PS D:\CPP> |
```

# Installing code runner in VS Code

- Install the **Code Runner** extension by searching for '**Code Runner**' in the Extensions view (Ctrl+Shift+X).

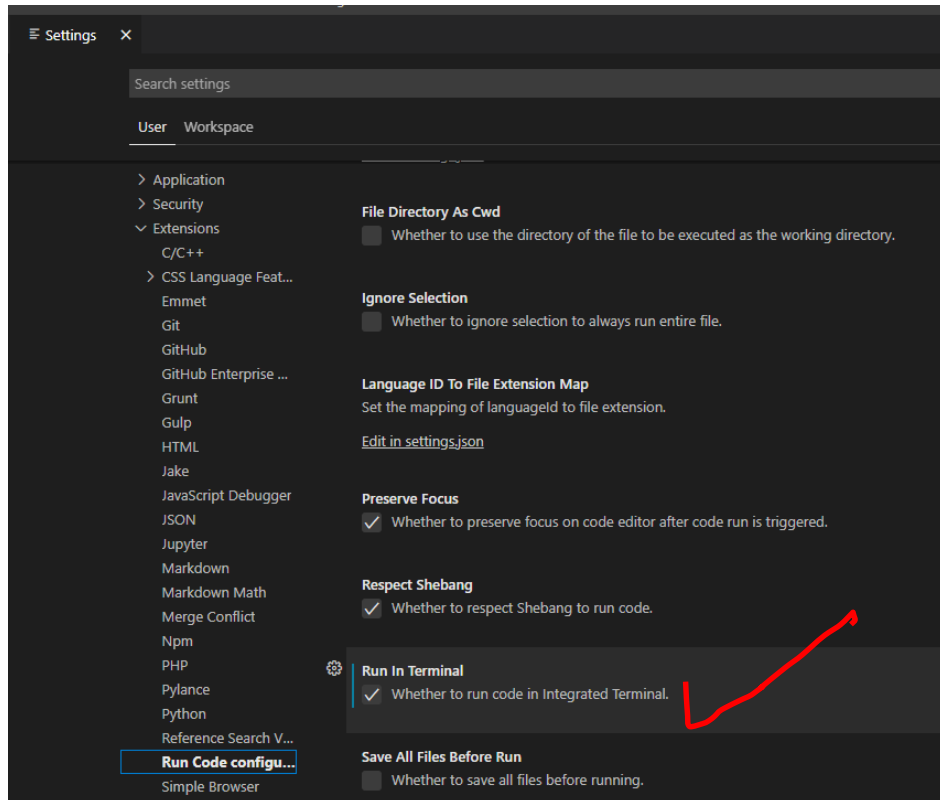


- After installation you will get code runner :



# For command line input

- Enable Run in Terminal in settings
  - ❑ Select File/preferences/settings
  - ❑ From settings select extensions/code Runner configuration/ check **run in terminal**



# Example

```
example.cpp > ...
1
2 #include<iostream>
3
4 int main()
5
6 {
7     int num1,num2;
8     std::cout<< "Enter Number 1 : ";
9     std::cin>>num1;
10    std::cout<< "Enter Number 2: ";
11    std::cin>>num2;
12    std::cout<<"The sum is : "<<num1 + num2;
13
14 }
```

If Run in Terminal  
not enabled

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
[Running] cd "d:\CPP\" && g++ example.cpp -o example && "d:\CPP\"example
Enter Number 1 : |
```

After Run in Terminal  
Is enabled

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS D:\CPP> cd "d:\CPP\" ; if ($?) { g++ example.cpp -o example } ; if ($?) { .\example }
Enter Number 1 : 20
Enter Number 2: 60
The sum is : 80
PS D:\CPP> |
```

---

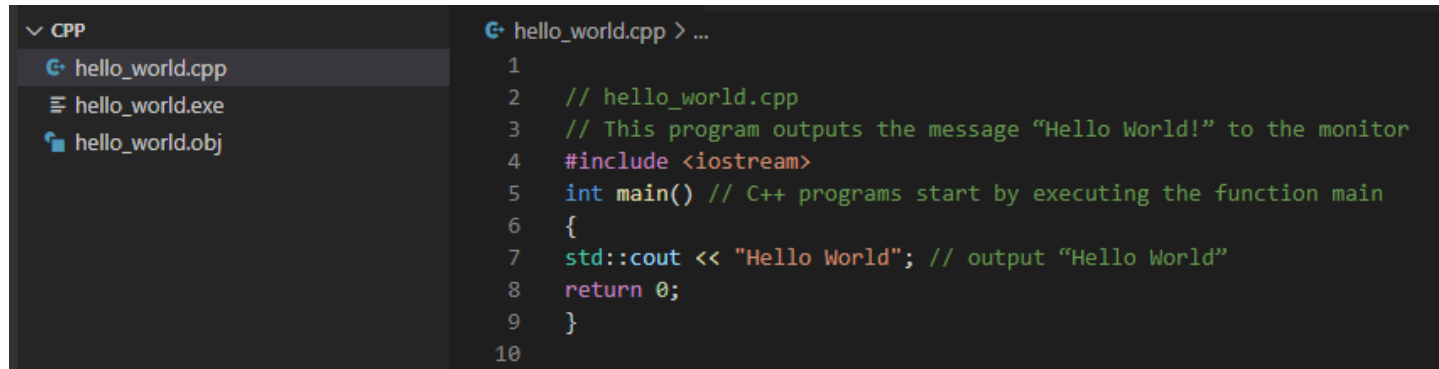
## Visual Studio Code on macOS

- <https://code.visualstudio.com/docs/setup/mac>



# Compilation and Linking Demo (Lab 1)

Write the following code in the editor:



```
1  // hello_world.cpp
2  // This program outputs the message "Hello World!" to the monitor
3  #include <iostream>
4  int main() // C++ programs start by executing the function main
5  {
6      std::cout << "Hello World"; // output "Hello World"
7      return 0;
8  }
```

The screenshot shows a code editor with a dark theme. On the left, a file explorer pane shows a project named 'CPP' containing three files: 'hello\_world.cpp' (selected), 'hello\_world.exe', and 'hello\_world.obj'. The main editor area displays the source code for 'hello\_world.cpp' with line numbers 1 through 10. The code is a simple C++ program that prints 'Hello World' to the console.

# Use g++ or clang compiler

Command Prompt

```
2 Dir(s) 155,738,722,304 bytes free

D:\CPP>g++ --version
g++ (GCC) 11.2.0
Copyright (C) 2021 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.

D:\CPP>g++ -c -o hello_world.obj hello_world.cpp

D:\CPP>g++ -o hello_world.exe hello_world.obj

D:\CPP>hello_world
Hello C++
D:\CPP>dir
Volume in drive D is Data
Volume Serial Number is A8AC-BCB9

Directory of D:\CPP

02/04/2022 05:10 PM <DIR> .
02/04/2022 05:10 PM <DIR> ..
02/04/2022 12:07 PM          351 hello_world.cpp
02/04/2022 05:10 PM    2,965,139 hello_world.exe
02/04/2022 05:10 PM      1,839 hello_world.obj
                3 File(s)      2,967,329 bytes
                2 Dir(s) 155,735,752,704 bytes free
```

Developer Command Prompt for VS 2022

```
D:\CPP>clang --version
clang version 12.0.0
Target: i686-pc-windows-msvc
Thread model: posix
InstalledDir: C:\Program Files\Microsoft Visual Studio\2022\Community\VC\Tools\Llvm\bin

D:\CPP>clang -c -o hello_world.obj hello_world.cpp

D:\CPP>clang -o hello_world.exe hello_world.obj

D:\CPP>hello_world.exe
Hello C++
D:\CPP>dir
Volume in drive D is Data
Volume Serial Number is A8AC-BCB9

Directory of D:\CPP

02/10/2022 04:12 PM <DIR> .
02/10/2022 04:12 PM <DIR> ..
02/09/2022 05:36 PM          351 hello_world.cpp
02/10/2022 04:12 PM    181,760 hello_world.exe
02/10/2022 04:12 PM      62,314 hello_world.obj
                3 File(s)      244,425 bytes
                2 Dir(s) 155,707,822,080 bytes free

D:\CPP>
```

```
PS D:\CPP>
PS D:\CPP> g++ -c -o hello_world.obj hello_world.cpp
PS D:\CPP> g++ -o hello_world hello_world.obj
PS D:\CPP> ./hello_world
Hello World
PS D:\CPP> |
```