

C Programming Environment Setup on Windows

Using Visual Studio Code and GCC

Prepared by: Chunghyun Lim (Class of 2019)

Table of Contents

1. Installing VS Code and Extensions

1. Installing VS Code

2. Installing C/C++ extensions for VS Code

3. Additional Settings (Optional)

2. Installing the GCC Compiler

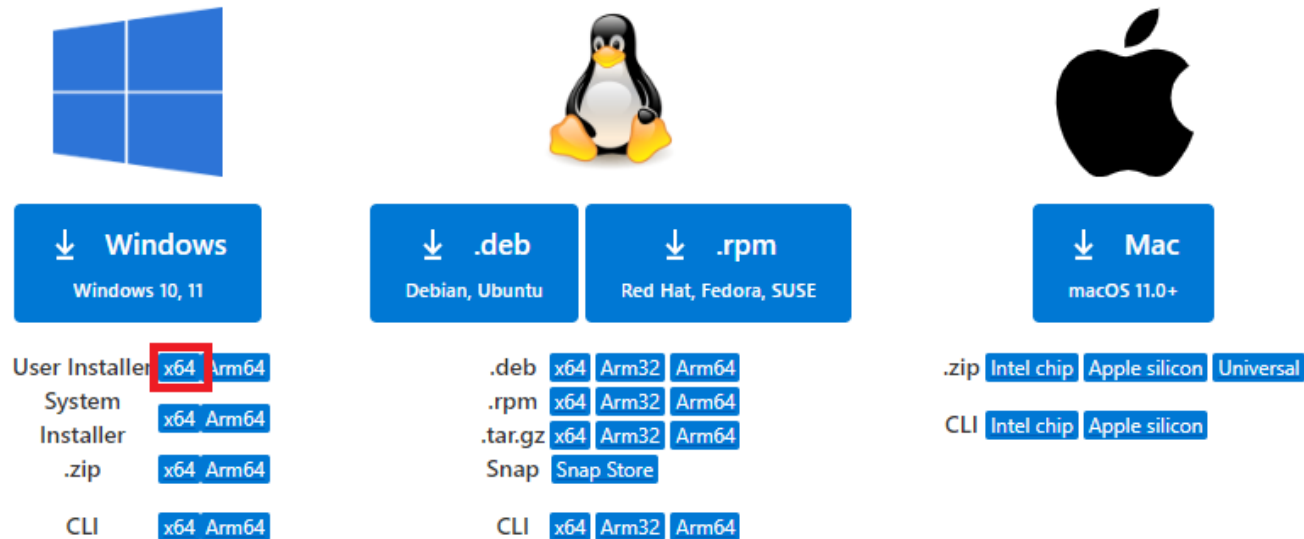
3. Writing and Running Your First C Program

1.1. Installing VS Code(1/3)

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

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

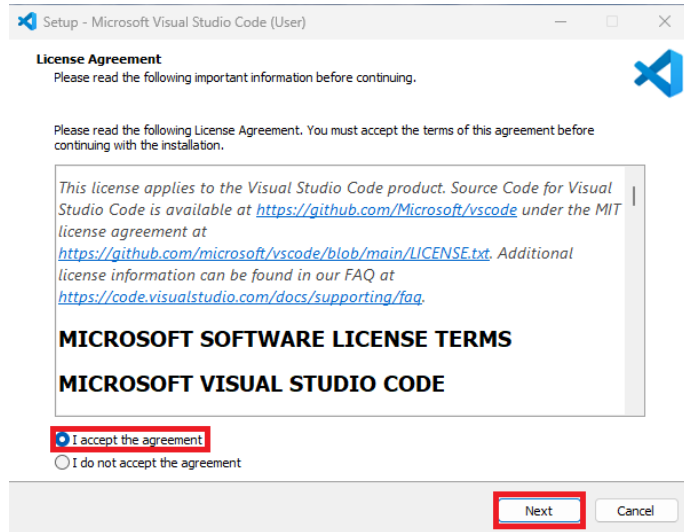


The image shows the download page for Visual Studio Code. It features three main sections: Windows, Linux, and Mac. Each section has a logo at the top, a download button, and a list of available installers. The Windows section has a blue Windows logo and a button labeled 'Windows' with a download icon. Below it, a list of installers is shown: 'User Installer' (with 'x64' highlighted in a red box), 'System Installer', '.zip', and 'CLI'. The Linux section has a Tux penguin logo and two buttons: '.deb' (for Debian, Ubuntu) and '.rpm' (for Red Hat, Fedora, SUSE). Below these, a list of installers is shown: '.deb', '.rpm', '.tar.gz', 'Snap', and 'CLI'. The Mac section has an Apple logo and a button labeled 'Mac' with a download icon. Below it, a list of installers is shown: '.zip' and 'CLI'.

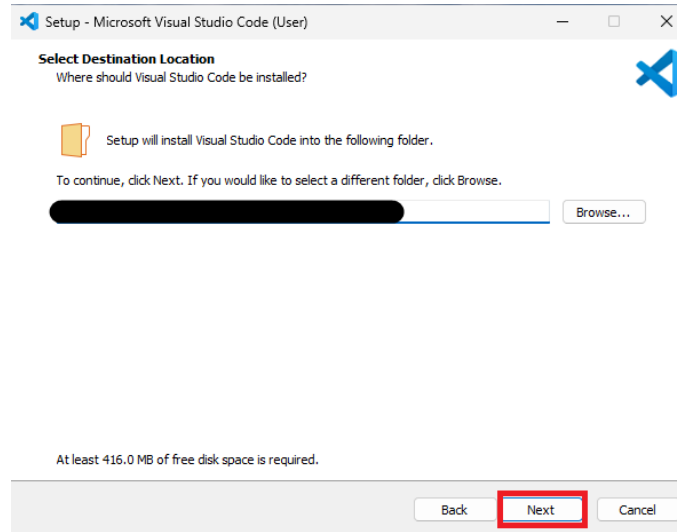
Platform	OS/Architecture	Available Installers
Windows	Windows 10, 11	User Installer (x64, Arm64), System Installer (x64, Arm64), .zip (x64, Arm64), CLI (x64, Arm64)
	Linux (Debian, Ubuntu)	.deb (x64, Arm32, Arm64)
		.rpm (x64, Arm32, Arm64)
	Linux (Red Hat, Fedora, SUSE)	.tar.gz (x64, Arm32, Arm64)
		Snap (Snap Store), CLI (x64, Arm32, Arm64)
Mac	macOS 11.0+	.zip (Intel chip, Apple silicon, Universal), CLI (Intel chip, Apple silicon)

Click User Installer (x64) → Run the downloaded file to begin installation

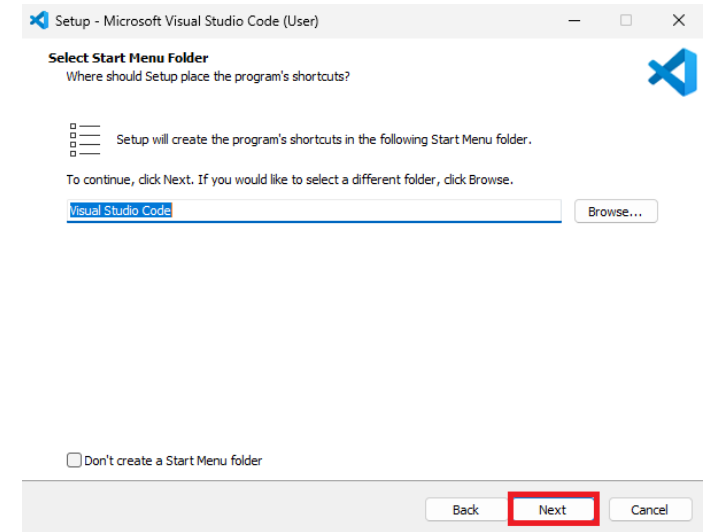
1.1. Installing VS Code(2/3)



- Check I accept the agreement.
- Click Next to continue.

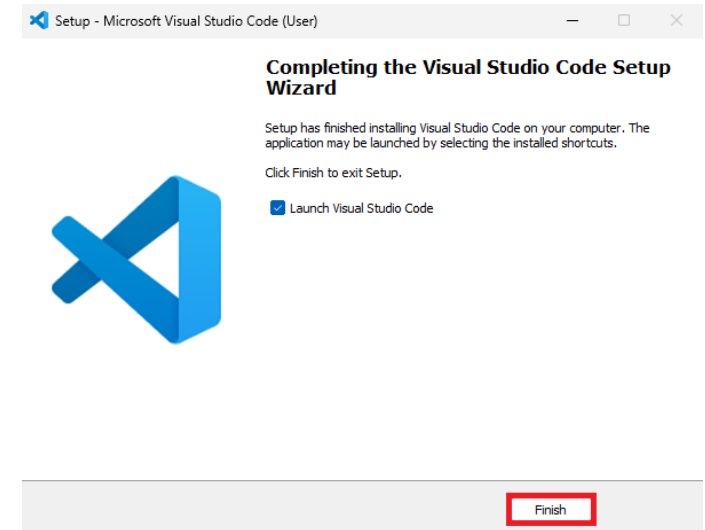
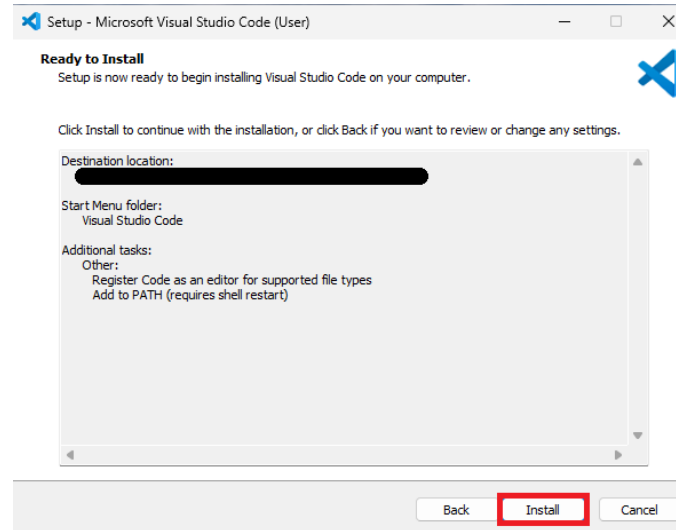
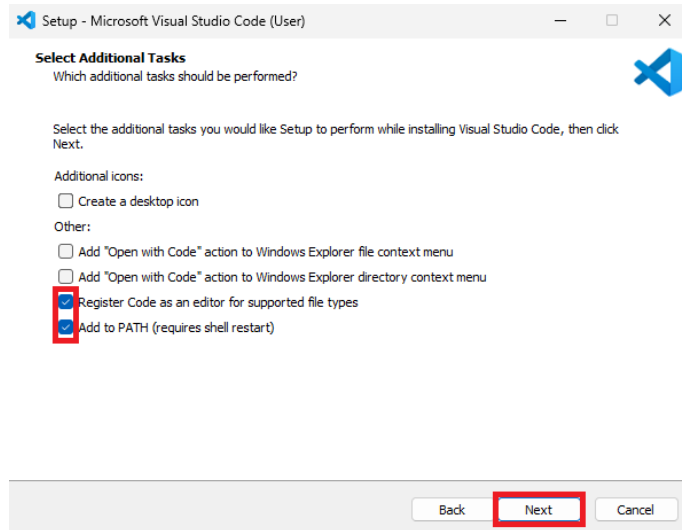


- Click Next to continue.



- Do not check Don't create a Start Menu folder.
- Click Next to continue

1.1. Installing VS Code(3/3)

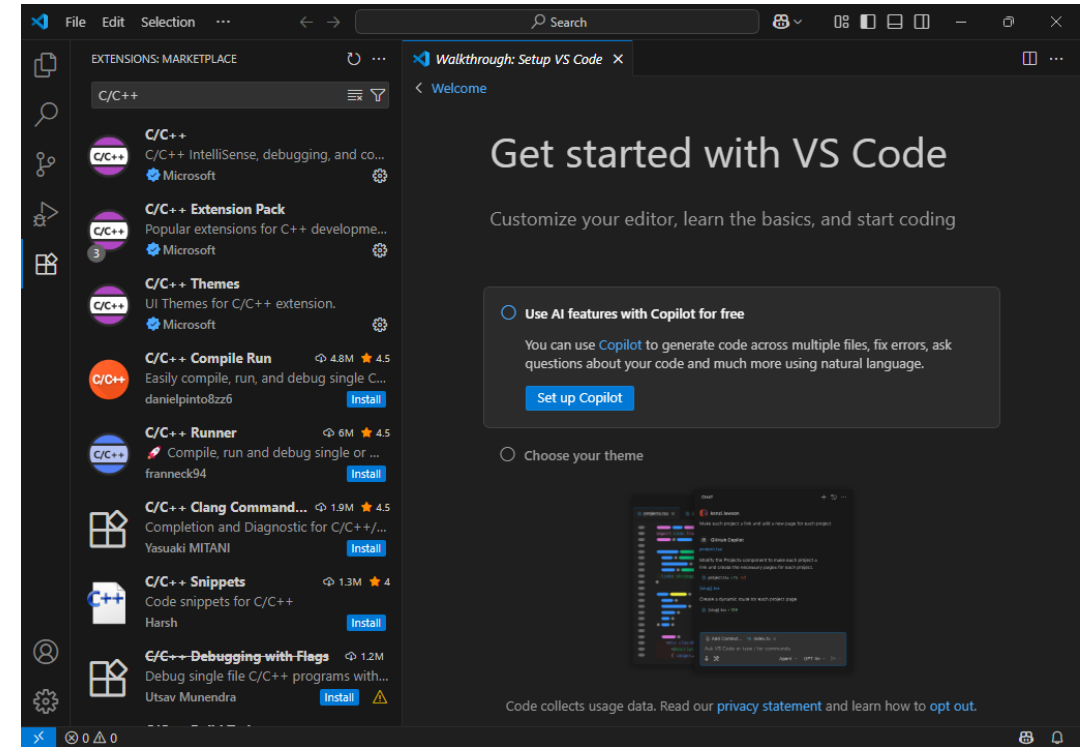
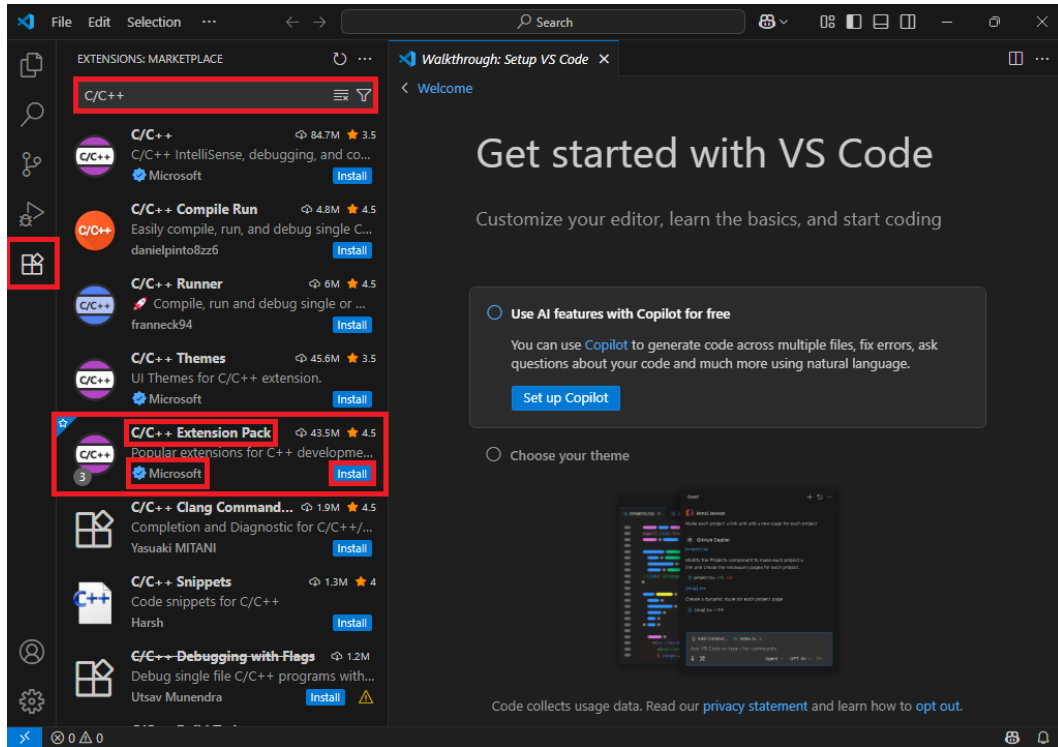


- Check Register Code as an editor for supported file types.
- Check Add to PATH.
- Click Next to continue.

- Click Install to begin the installation.

- Check Launch Visual Studio Code.
- Click Finish to complete the installation.

1.2. Installing C/C++ Extensions for VS Code

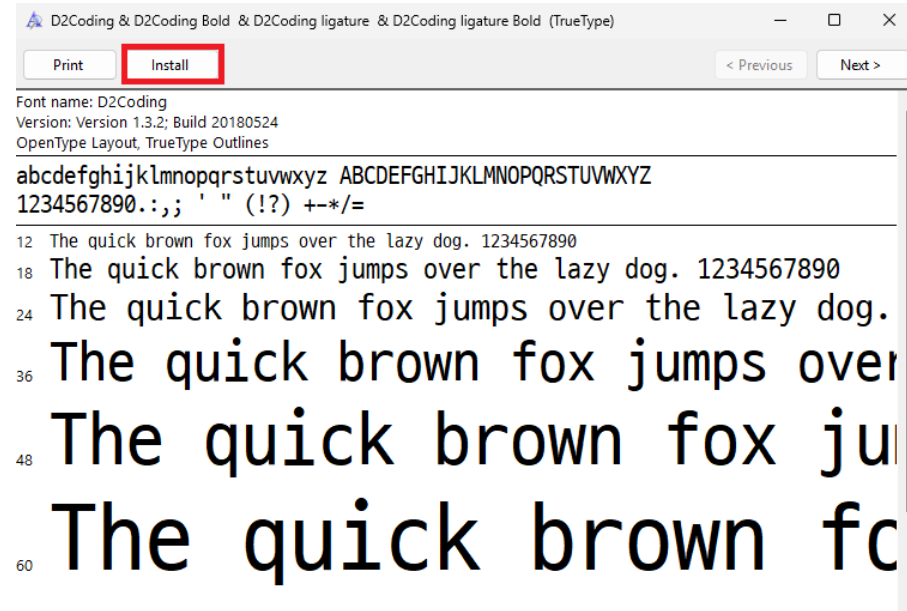
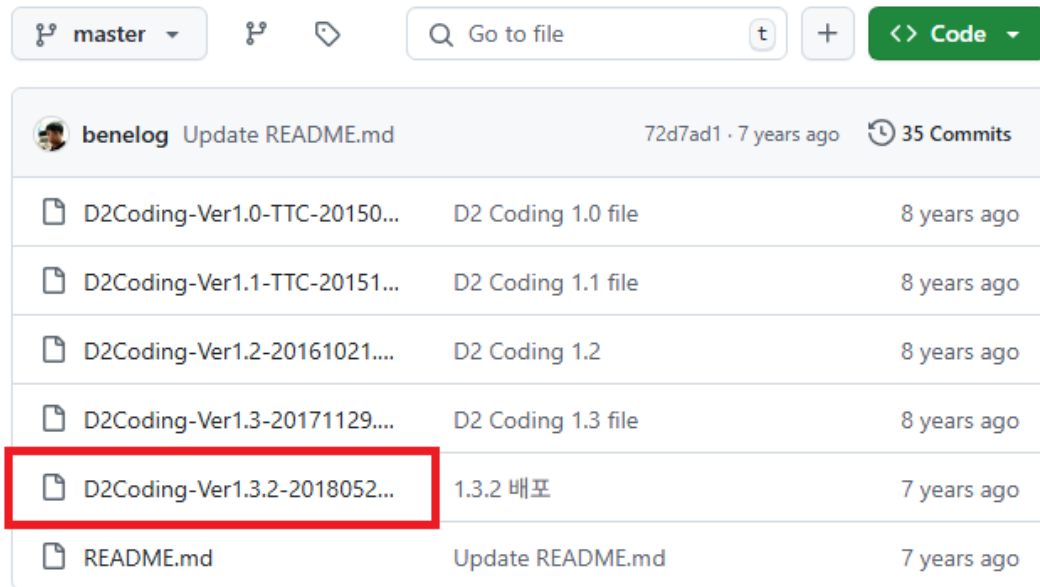


- Click the Extensions icon in the Activity Bar on the left.
- In the search bar, type C/C++.
- Find and install C/C++ Extension Pack by Microsoft.

- After installing, 4 extensions are automatically added:
C/C++, C/C++ Extension Pack, C/C++ Themes, CMake Tools

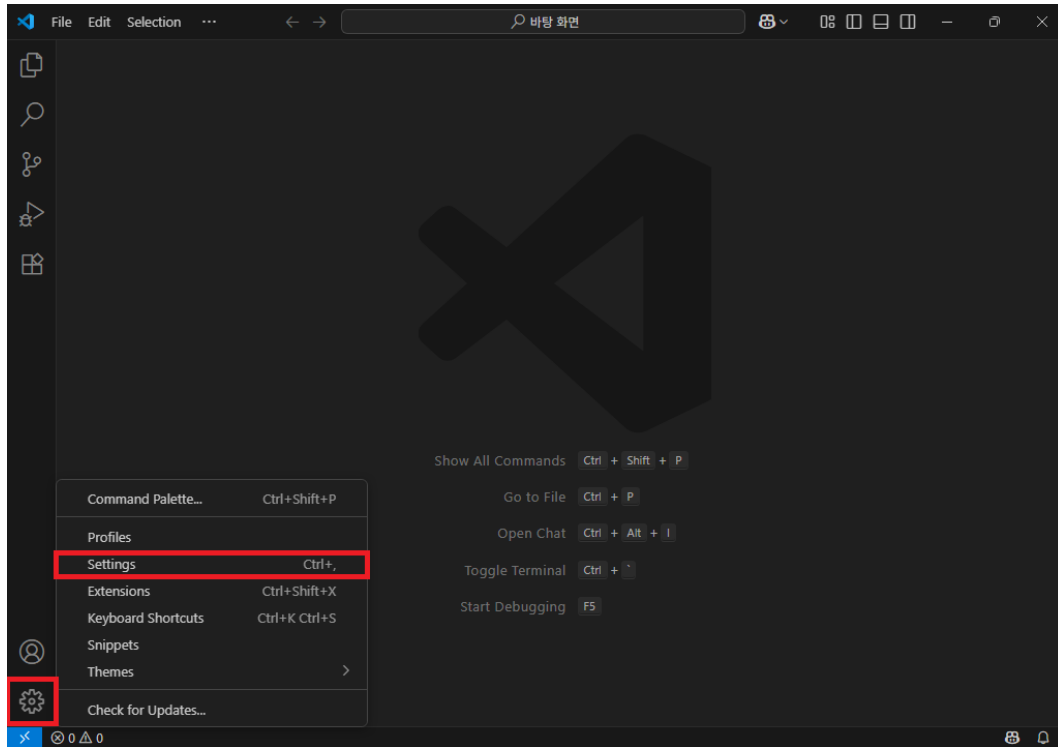
1.3. Additional Settings (Optional) – Fonts(1/2)

Download link: <https://github.com/naver/d2codingfont>

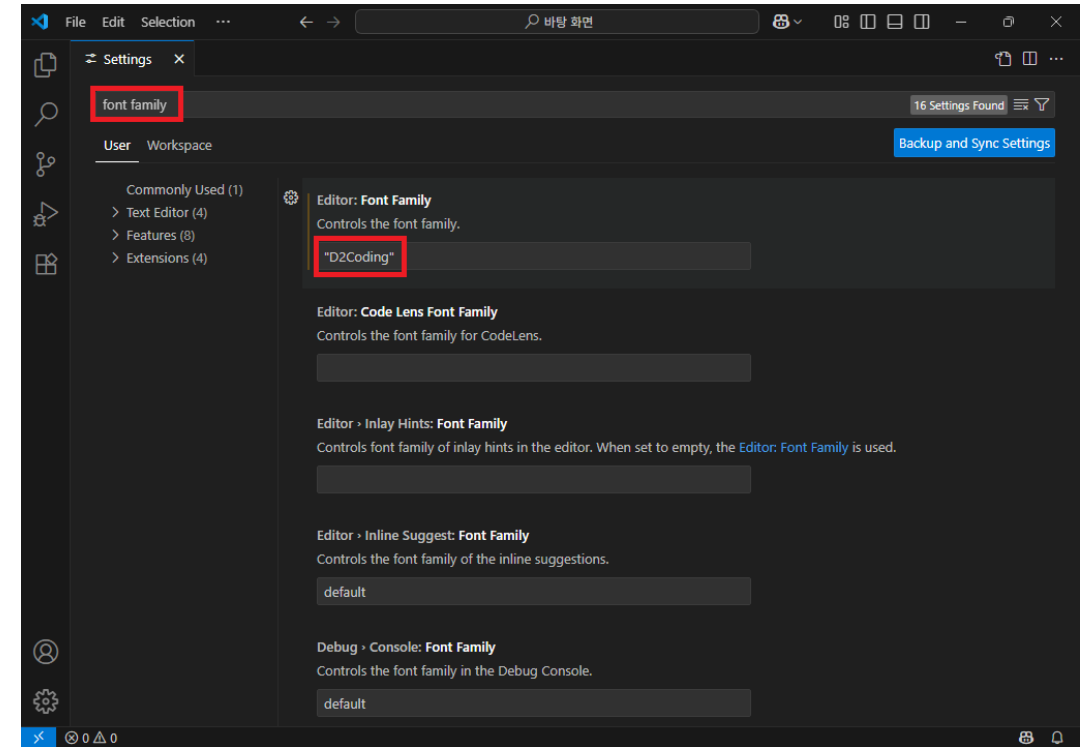


- Download the latest version.
- Extract the downloaded archive.
- Open the .ttc file inside the D2CodingAll.
- Click the Install button to install the font.

1.3. Additional Settings (Optional) – Fonts(2/2)

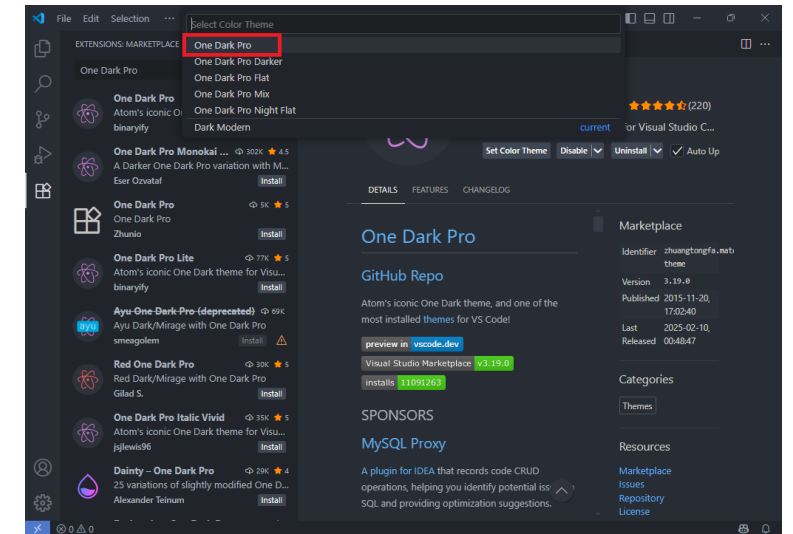
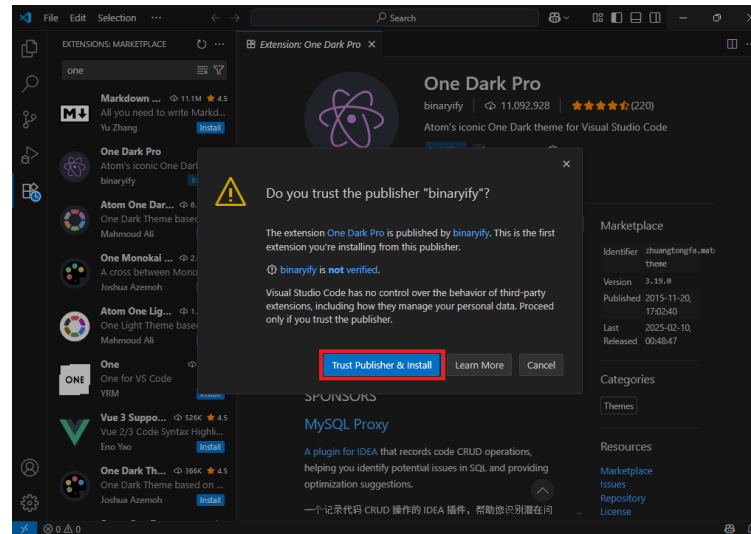
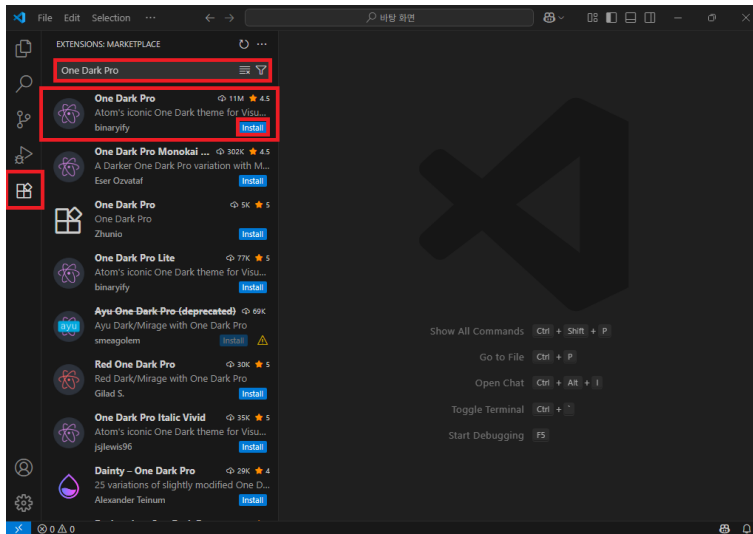


- Click the gear icon in the bottom-left corner to open the menu.
- From the menu, select Settings.



- In the search bar at the top, type font family.
- In Editor: Font Family, type "D2Coding" to set the editor font.

1.3. Additional Settings (Optional) – Themes



- Click the Extensions icon in the Activity Bar on the left.
- In the search bar, type One Dark Pro.
- Find and install One Dark Pro by binaryify.

- Click Trust Publisher & Install when prompted.
- This completes the theme installation.

- After installation, a list of themes appears.
- Select One Dark Pro to apply it.

2. Installing the GCC Compiler(1/4)

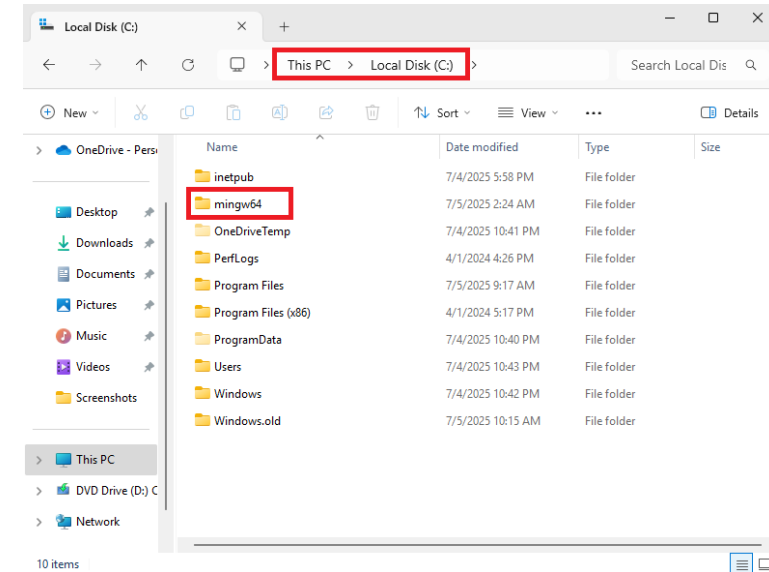
Download link: <https://winlibs.com>

Release versions

UCRT runtime

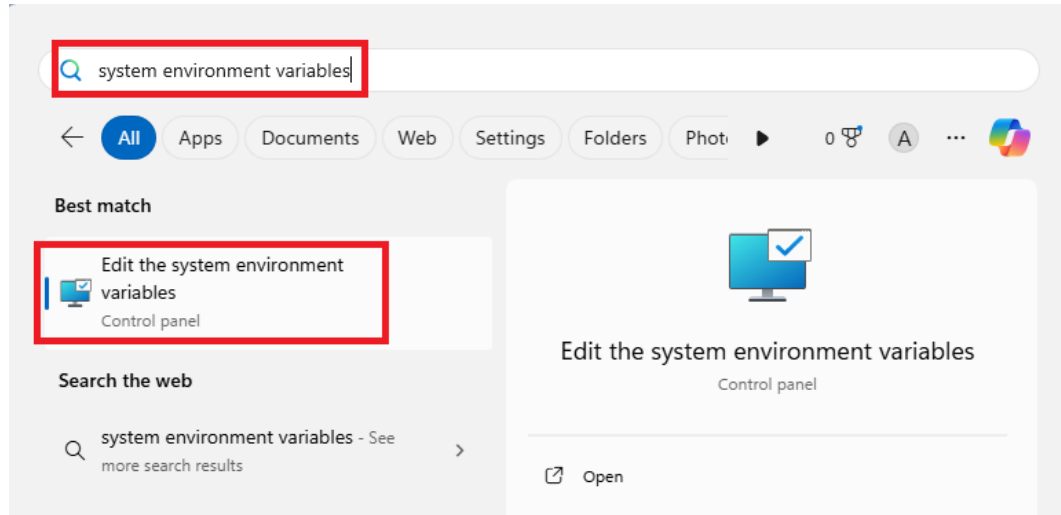
- GCC 15.1.0 (with **POSIX** threads) + MinGW-w64 13.0.0 UCRT - release 2 **(LATEST)**
 - Win32 (without LLVM/Clang/LLD/LLDB): [7-Zip archive*](#) | [Zip archive](#)
 - Win64 (without LLVM/Clang/LLD/LLDB): [7-Zip archive*](#) | [Zip archive](#)
- GCC 15.1.0 (with **POSIX** threads) + MinGW-w64 12.0.0 UCRT - release 1
 - Win32 (without LLVM/Clang/LLD/LLDB): [7-Zip archive*](#) | [Zip archive](#)
 - Win64 (without LLVM/Clang/LLD/LLDB): [7-Zip archive*](#) | [Zip archive](#)

- Click on Zip archive (Win64, UCRT) to download the latest GCC version.
Recommended: version with POSIX threads
- Extract the downloaded archive.

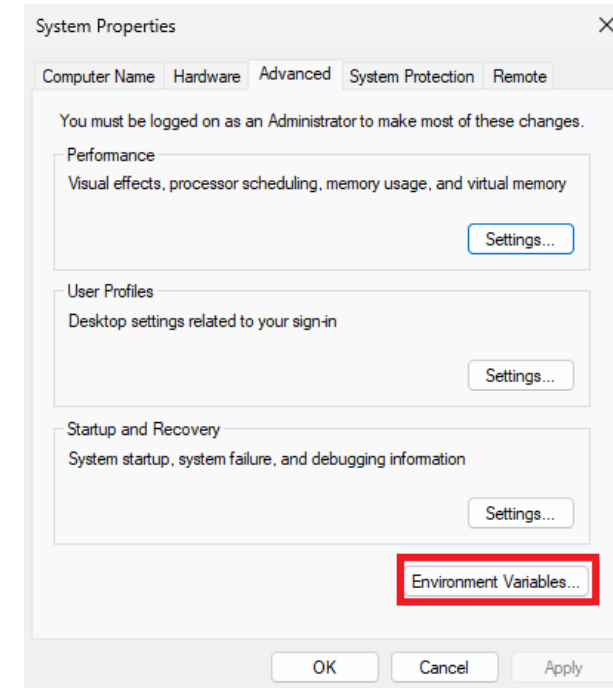


- Move the root-level mingw64 folder to the C: drive.
- Make sure the final path is exactly C:\mingw64.

2. Installing the GCC Compiler(2/4)

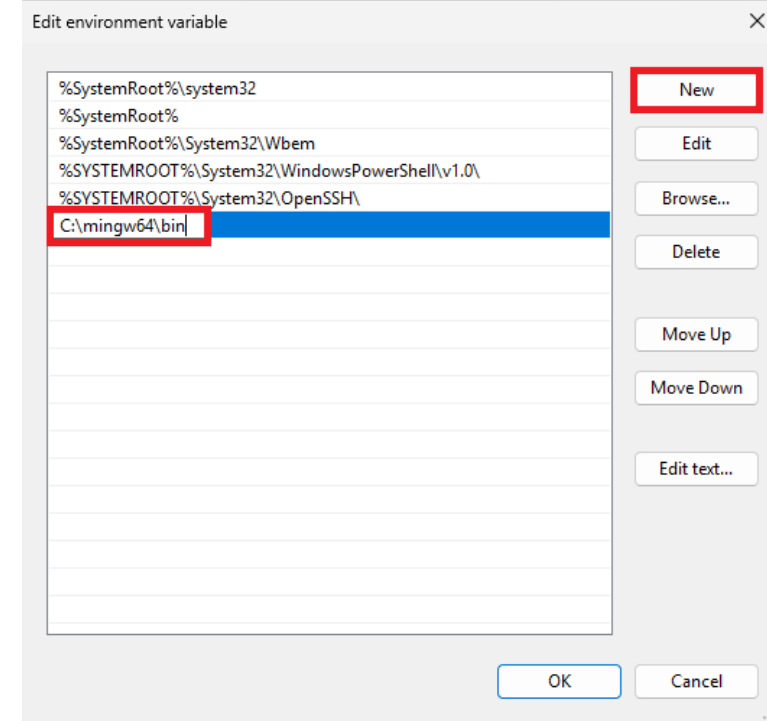
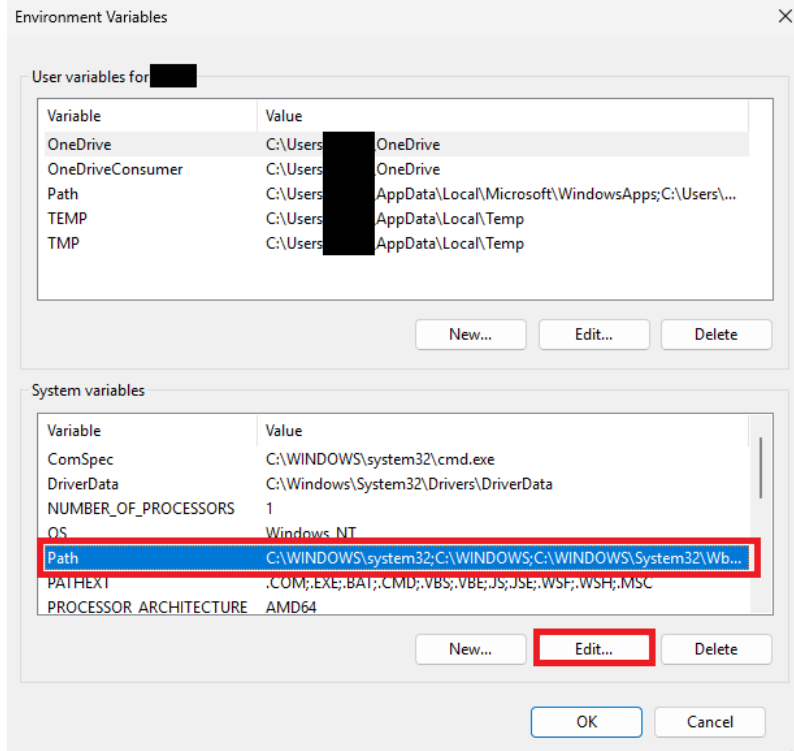


- Press the Windows key and type system environment variables.
- Select "Edit the system environment variables" from the search results.



- Click the "Environment Variables..." button at the bottom.

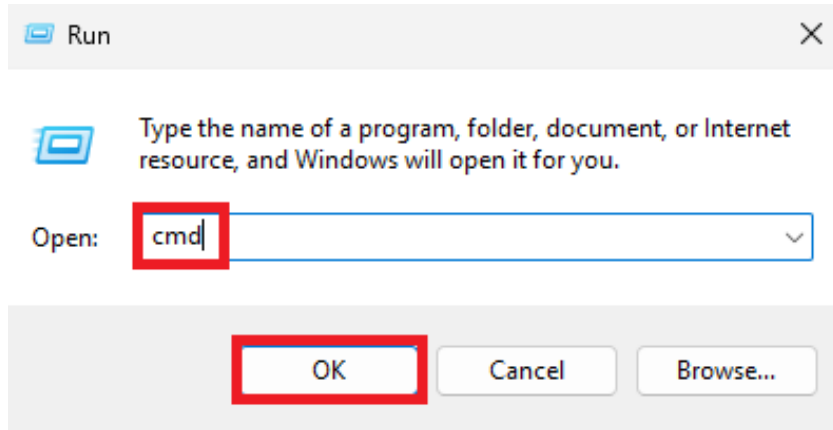
2. Installing the GCC Compiler(3/4)



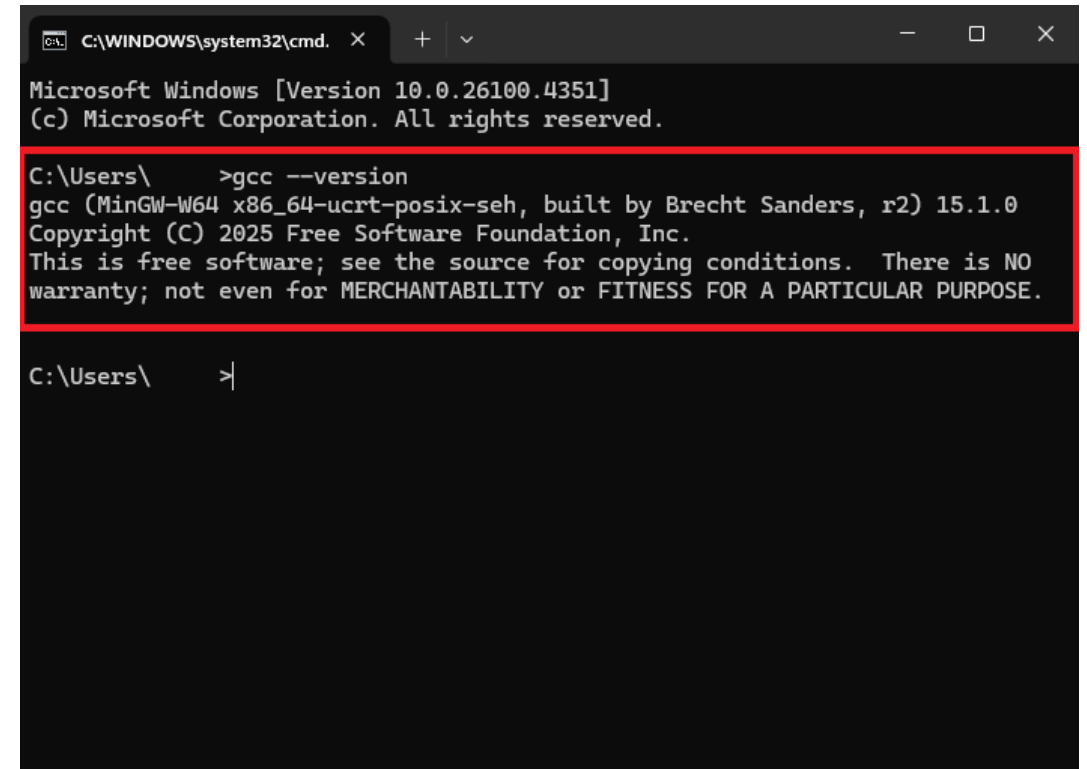
- Under System variables, find and select the Path variable.
- Click "Edit..." to open the list of environment paths.

- Click "New" and add the following path: C:\mingw64\bin
- This allows you to run gcc and related commands from any terminal.
- Click OK on all open dialogs to save the changes and close the windows.

2. Installing the GCC Compiler(4/4)

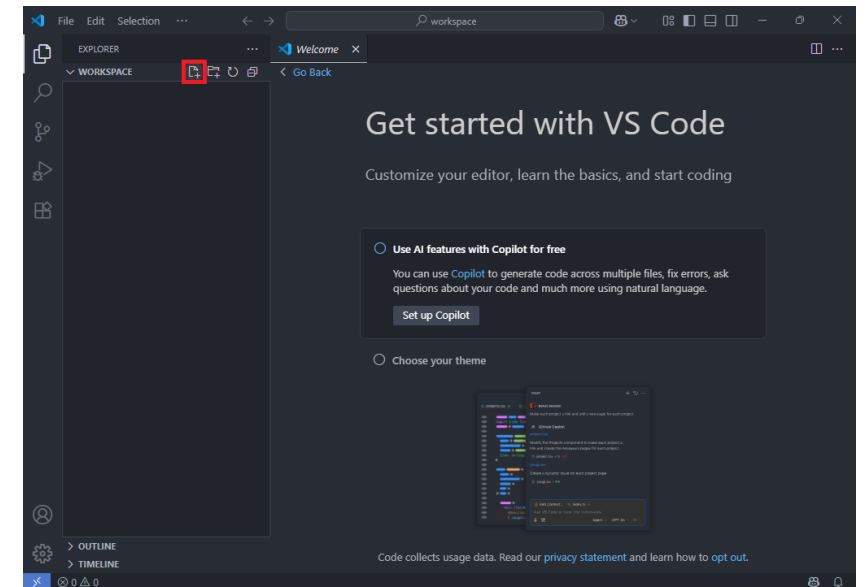
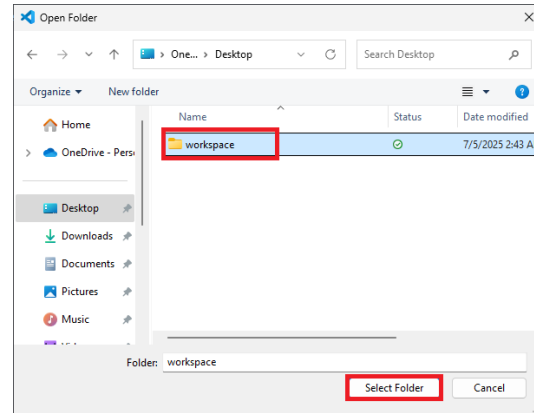
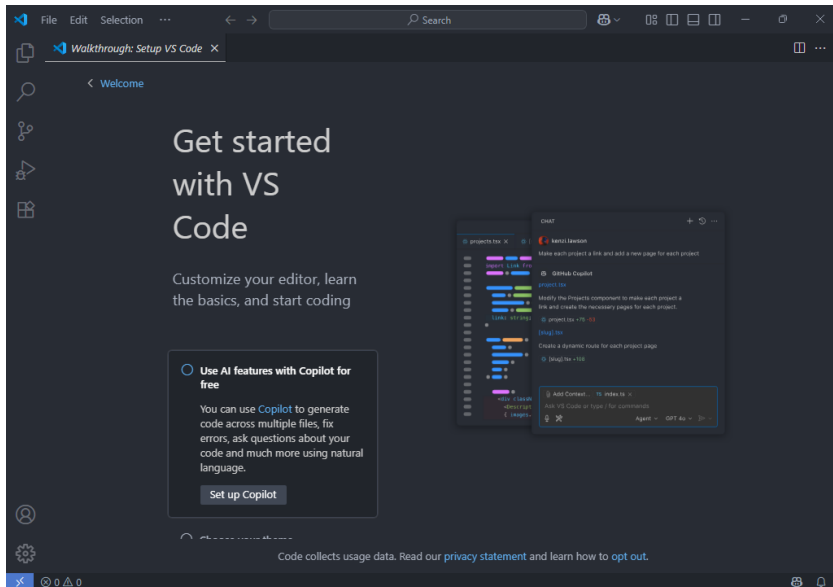


- Press Windows + R, type cmd in the Run dialog box, and click OK

A screenshot of a Windows Command Prompt window. The title bar shows 'C:\WINDOWS\system32\cmd.' and standard window controls. The window content shows the Microsoft Windows version (10.0.26100.4351) and copyright information. Below this, the command `>gcc --version` has been entered and executed. The output is displayed in a monospaced font and is highlighted with a red rectangle. The output text is: `gcc (MinGW-W64 x86_64-ucrt-posix-seh, built by Brecht Sanders, r2) 15.1.0`, `Copyright (C) 2025 Free Software Foundation, Inc.`, and `This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.` Below the output, the prompt `C:\Users\ >` is visible.

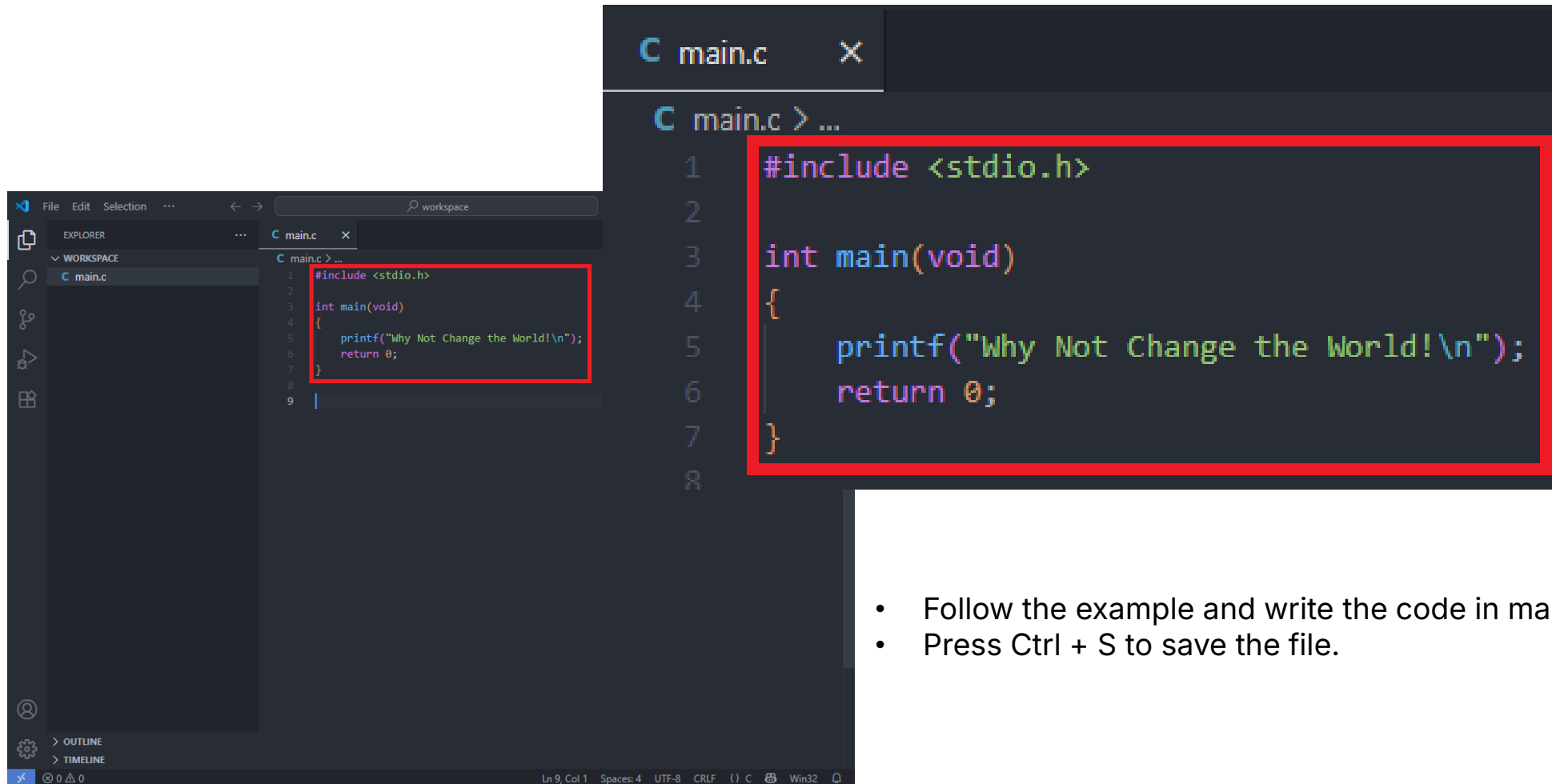
- To verify the setup, open Command Prompt and run `gcc --version`.
- The version number should be displayed without any errors.

3. Writing and Running Your First C Program(1/3)



- Launch Visual Studio Code.
- Press Ctrl + K, then O to open folder you want to use.
- Create a folder and click Select Folder.
- Click the New File icon and create a file named main.c.

3. Writing and Running Your First C Program(2/3)



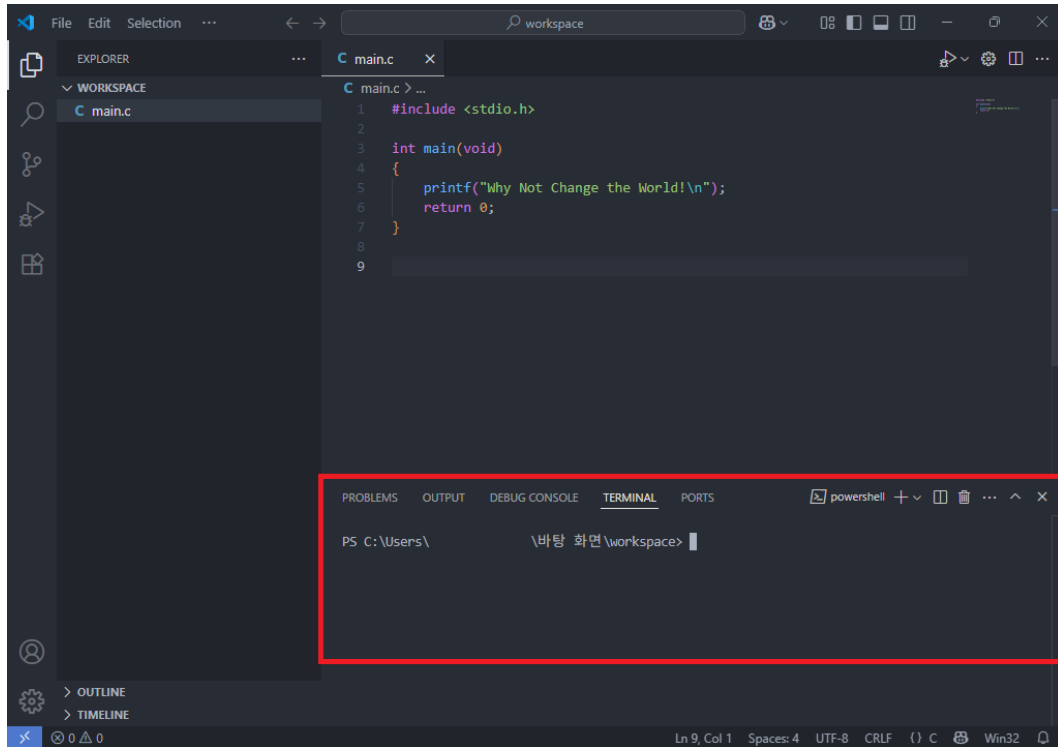
The image shows a screenshot of the Visual Studio Code editor. On the left, the Explorer sidebar shows a workspace with a file named `main.c`. The main editor window displays the code for `main.c`. The code is as follows:

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     printf("Why Not Change the World!\n");
6     return 0;
7 }
```

The code is highlighted with a red rectangular box. The status bar at the bottom indicates the current position is Line 9, Column 1, with 4 spaces, UTF-8 encoding, CRLF line endings, and a Win32 file system.

- Follow the example and write the code in `main.c`.
- Press `Ctrl + S` to save the file.

3. Writing and Running Your First C Program(3/3)

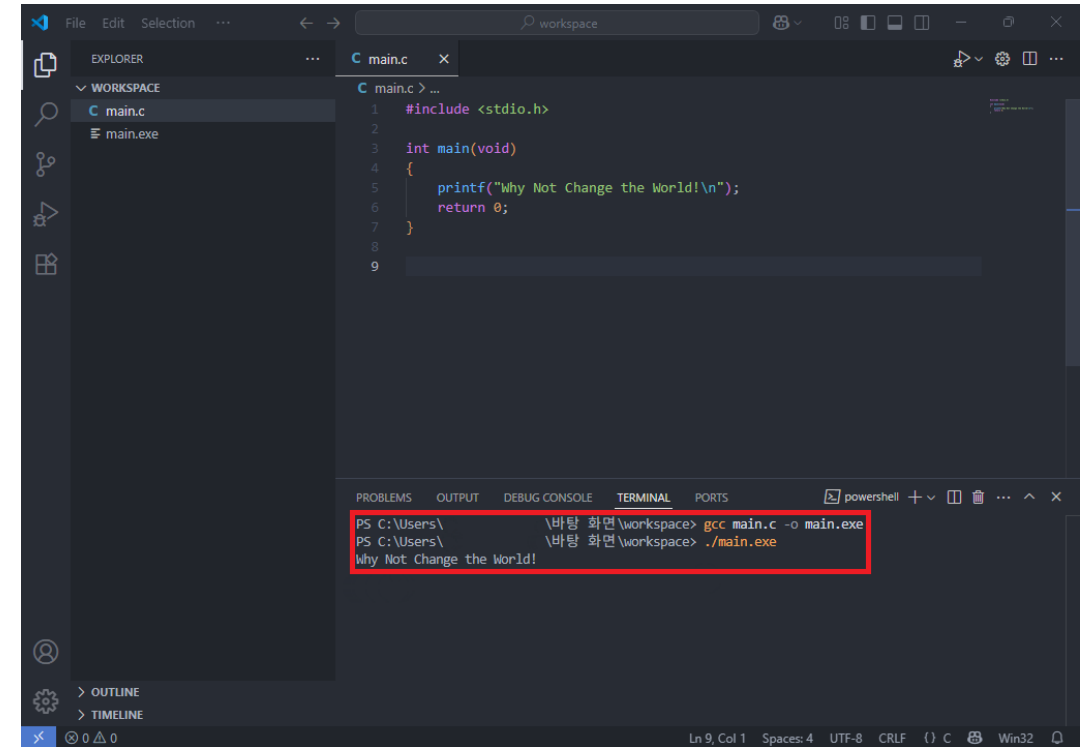


The screenshot shows the Visual Studio Code editor with a C program named `main.c` open. The code is as follows:

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     printf("Why Not Change the World!\n");
6     return 0;
7 }
8
9
```

The terminal panel at the bottom is highlighted with a red box. It shows the command prompt at `PS C:\Users\바탕 화면\workspace>`.

- Press Ctrl + ` to open the integrated terminal at the bottom.



The screenshot shows the Visual Studio Code editor with the same C program `main.c`. The terminal panel at the bottom, also highlighted with a red box, shows the following commands and output:

```
PS C:\Users\바탕 화면\workspace> gcc main.c -o main.exe
PS C:\Users\바탕 화면\workspace> ./main.exe
Why Not Change the World!
```

- In the terminal, type: `gcc main.c -o main.exe`
- In the terminal, type: `./main.exe`
- Seeing the printed message means the setup is complete.

Thank you

Wishing you an enjoyable and meaningful learning experience.

May you grow into an engineer who learns with joy and shares with others.

For inquiries, contact: potterLim0808@gmail.com

