

Getting Started with mcpp

To communicate with Minecraft using C++, you must set up an environment using a Minecraft server that runs on your machine and the C++ library mcpp (Minecraft++), which communicates with that server.

The Minecraft Spigot server in conjunction with mcpp is capable of, for example, placing or destroying a block in the world.

Below, you will find instructions on how to:

1. Set up a Linux environment via **Windows Subsystem for Linux Version 2** (a.k.a. WSL 2, only required for Windows users).
2. Download and run a Minecraft Spigot server with an installation of the Java Runtime.
3. Install the required version of the Minecraft game itself (1.19.4) and join the Spigot server.
4. Install the necessary C++ tools (the g++ compiler and the mcpp library).
5. Test the full setup via a simple "Hello, World!" program that sends a chat message to Minecraft.

Purchasing Minecraft Java Edition

If you don't own a Minecraft Java Edition copy, please see the purchasing instructions [here](#) before continuing with this guide. Note that even if you are using WSL to run the Minecraft server, you can still install the Windows Minecraft client.

1. Windows Prerequisites

Note: If you are not a Windows user, go to **Section 2: Installing and running a Minecraft server**.

Since your programs are expected to run on Linux/UNIX machines, Windows users must install WSL (Windows Subsystem for Linux) for development. This is like having a Linux virtual machine, but it runs alongside your OS with a minimal footprint.

Installing WSL

Follow the instructions linked below to install WSL 2 on your machine.

Important note: When selecting a distribution, choose Ubuntu **22.04**, i.e., `wsl --install -d Ubuntu-22.04``.

<https://learn.microsoft.com/en-us/windows/wsl/install> 

Notes:

- The version of Ubuntu (22.04) matters! If you have an earlier distribution installed, you must upgrade it. See the top answer [here](#) ➞.
- If you already have an older WSL 1 distribution installed, upgrade it. If you're unsure which version is installed, open Windows Powershell and run the following command:

```
wsl -l -v
```

If this shows that you have version 1 installed, you can upgrade to version 2 by running:

```
wsl --set-version Ubuntu 2
```

Working with WSL

As a Windows user, whenever you want to test your C++ code or start a Minecraft server, you must do so via WSL.

After installing WSL 2, you can open a terminal by clicking the start button, typing "Ubuntu", then clicking on the app link that appears. You can navigate to your Windows C drive by running ``cd /mnt/c/``, and so on for other subfolders and/or drives.

However, we recommend you work in the WSL/Linux filesystem and store your code in your home directory (see [this page](#) ➞ for a basic intro, albeit we will cover these concepts in class). For instance, you may create a subfolder for your PS2 work in your home directory, e.g., `/home/username/PS2` (where **username** will be replaced by your username under WSL).

VSCode and WSL

It is tiresome to open WSL terminal windows repeatedly; integrating VSCode with WSL will make you more productive. Instructions can be found here:

<https://code.visualstudio.com/docs/remote/wsl> ➞. We will show this works in class; however, if you would like a comprehensive introduction to installing WSL, how that integrates into Windows and how to make VS Code work in WSL, please review [this excellent video](#) ➞.

2. Installing and running a Minecraft server

Note: From this point onward, all instructions referring to Linux/UNIX also refer to Windows users running WSL.

You must run a Minecraft server on your local machine for assignment development. Instructions on how to do so are provided below.

Spigot server download

Follow the steps below in your Linux/Mac terminal:

```
cd ~ && mkdir PS2
cd PS2
git clone https://github.com/rozukke/minecraft_tools.git
```

Install the Java Runtime Environment (JRE)

Before you can run the Server, you must have a Java Runtime Environment (JRE) installed.



WSL / Linux

Run the below commands **from inside a WSL terminal**. Complete the update and upgrade commands, as not doing so can cause several issues!

```
sudo apt update
sudo apt upgrade
sudo apt install openjdk-17-jdk
```

Mac

Note: You may not need to do this if you have completed Programming Studio 1 previously. Just verify the installation to check that you have Java installed.

Download the Java SE Development Kit [here](#) , go to [Java archive](#) . Select the correct version for your machine (ARM for M series Macs). Follow the installation prompts. Your installed version of JRE should either be **version 17 or 19**.

Verifying the installation

Check the version with ``java --version`` and verify it is **version 17 or 19**.

Running the Server

WSL / Linux

Open a WSL terminal window and navigate to your **/home/username/PS2** folder to access the tools you downloaded. By default, the path should end with "minecraft_tools".

For example, the Spigot server is at **/home/fabio/PS2/minecraft_tools** on my machine.

Change the current directory to that folder and then run ``./start_server.sh``.

If the `start_server` command fails due to a permission error, you may need to make the script executable using:

```
sudo chmod a+x start_server.sh // (if on Mac, it would be start_server.co
mmand)
```

You need to start it every time you want to interact with mcpp. Any time you want to rerun the Minecraft server, go to the Server folder and run ``./start_server.sh``.

Mac

Open a terminal window and navigate to your home folder ``cd ~`` and change the current directory to the "minecraft_tools" folder. By default, this should be in PS2/minecraft_tools, so you can run ``cd PS2/minecraft_tools``. Now run the following:

```
./start.command
```

Note: If `start.command` fails due to a permission error, you may need to make the script executable using: ``chmod u+x start.command``

You need to start it every time you want to interact with **mcpp**. Any time you want to rerun the Minecraft server, go to the Server folder and run ``./start.command``.

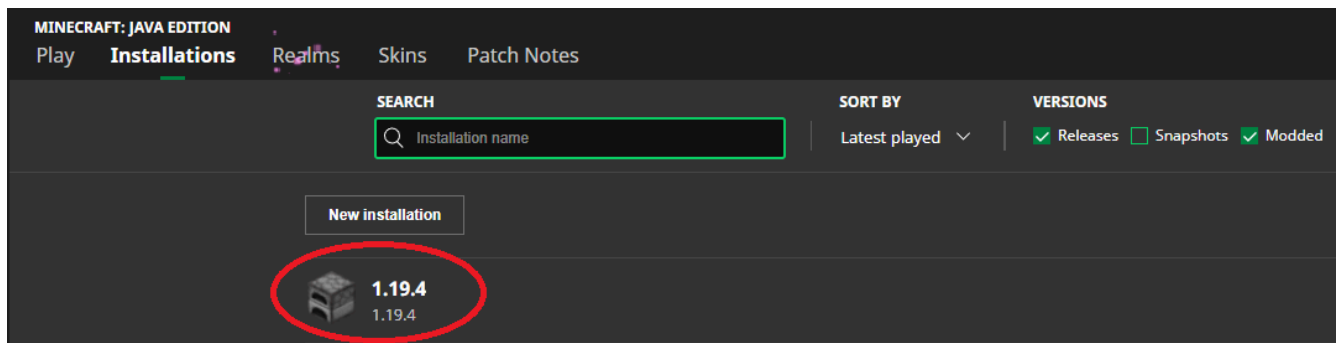
3. Joining the Server

Installing Minecraft 1.19.4

To avoid compatibility issues, it is necessary to use the **1.19.4** release of Minecraft.

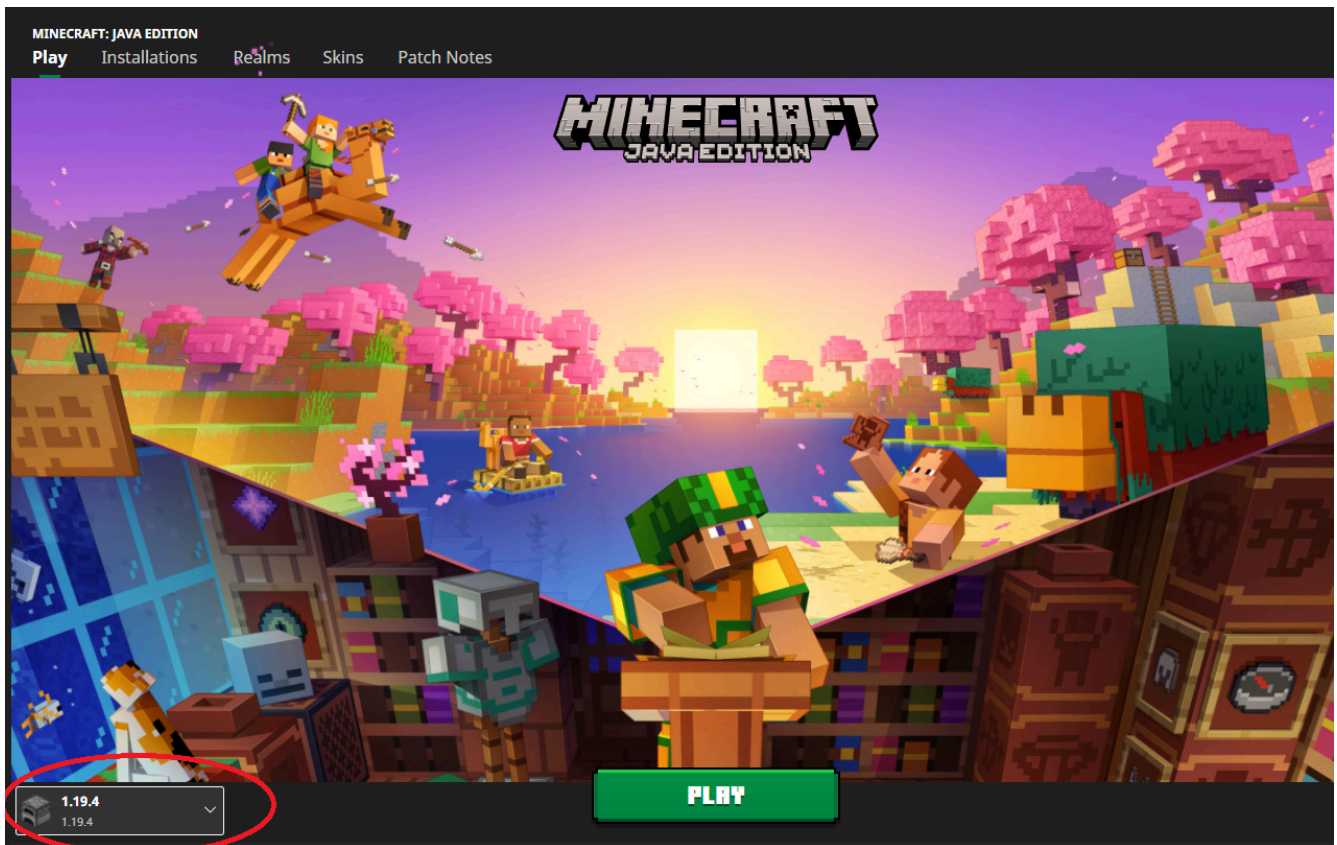
Run the Minecraft launcher, then select the **Installations** tab up the top. Select **New Installation**, name it "1.19.4", then select "release 1.19.4" from the dropdown and click **Create**.

If all goes well, you should be able to see it in your installations list afterwards:

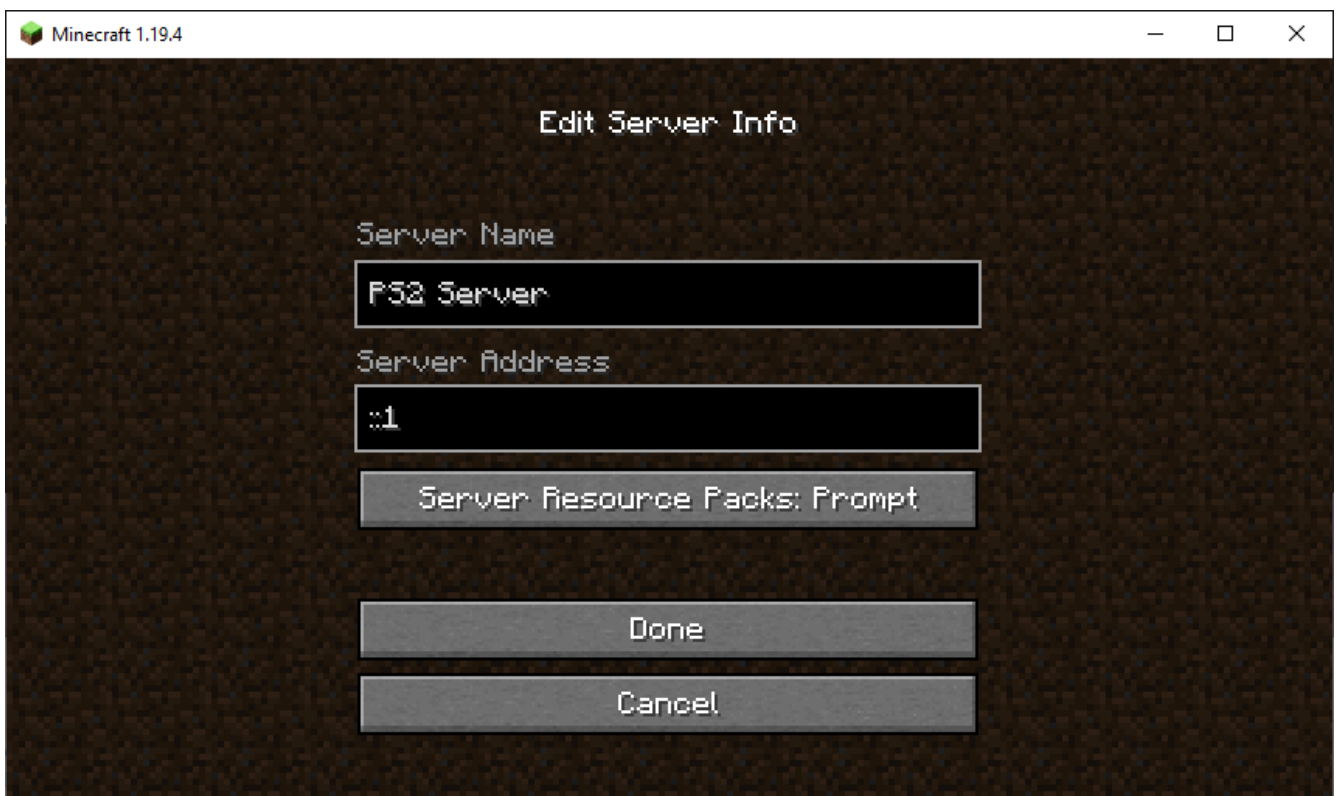


Connecting to the Server

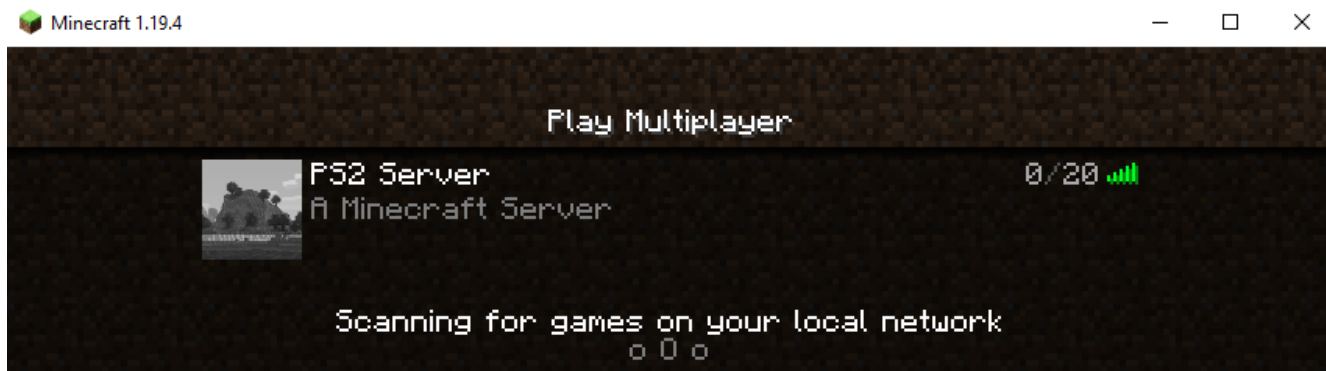
Now go back to the **Play** tab and ensure that you have the 1.19.4 release selected down at the bottom-left:



Click on the **Play** button. Once the game launches, click on **Multplayer**, then **Add Server**. Enter the details below, with server address "::<1", then click **Done**.



You should now be able to join a game by clicking **Join Server** or double-clicking the Server that appears:



4. Installing the C++ Compiler

WSL / Linux

To compile and run your code, install the **make** and **g++** packages from inside a WSL terminal using this command:

```
sudo apt-get install make g++
```

Mac

Install the compiler via the following command:

```
xcode-select --install
```

Verify Installation

To check that your compiler has been installed correctly, check the output of the command:

```
g++ -v
```

If g++ has been installed correctly, this should output its version info.

Installing mcpp

Linux/WSL:

Install **cmake**, then follow the shared steps below:

```
sudo apt install cmake
```

Mac:

First, check whether you have brew installed by running `brew`. If the command is not recognized, install it by running:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Next, install **cmake**:

```
brew install cmake
```

Shared steps for Linux/WSL/Mac:



Here are the shared steps for all systems:

```
cd ~ && mkdir PS2
cd PS2
git clone https://github.com/rozukey/mcpp.git && cd mcpp
cmake -B build && cd build
```

Then run the following commands as well. Don't worry if this returns an error like "file exists".

```
sudo mkdir /usr/local/lib
sudo make install
```

Further information

mcpp was developed by our student mentor Artemis, and you can browse the GitHub repo here: <https://github.com/rozukey/mcpp> . A documentation site is available via the repo. You may find this very helpful later, particularly when completing Assignment 3. [You can save the repository with stars](#)  in your GitHub account and in doing so help support one of your mentors.

5. A short test program

Create a file called `hello_minecraft.cpp` in a subfolder of your PS2 folder, e.g., `/home/username/PS2/hello`

Paste in the following contents:

```
#include <mcpp/mcpp.h>

int main() {

    mcpp::MinecraftConnection mc;

    mc.postToChat("Hello Minecraft!");

}
```

Open a terminal window (a WSL terminal if you are on Windows) and navigate to the folder where you created the file. Compile the program with the `-lmcpp` flag:

```
g++ -std=c++17 -Wall -o hello_minecraft hello_minecraft.cpp -lmcpp
```

Note #1: Make sure you are using **g++** and not **gcc** as that will cause compatibility issues.

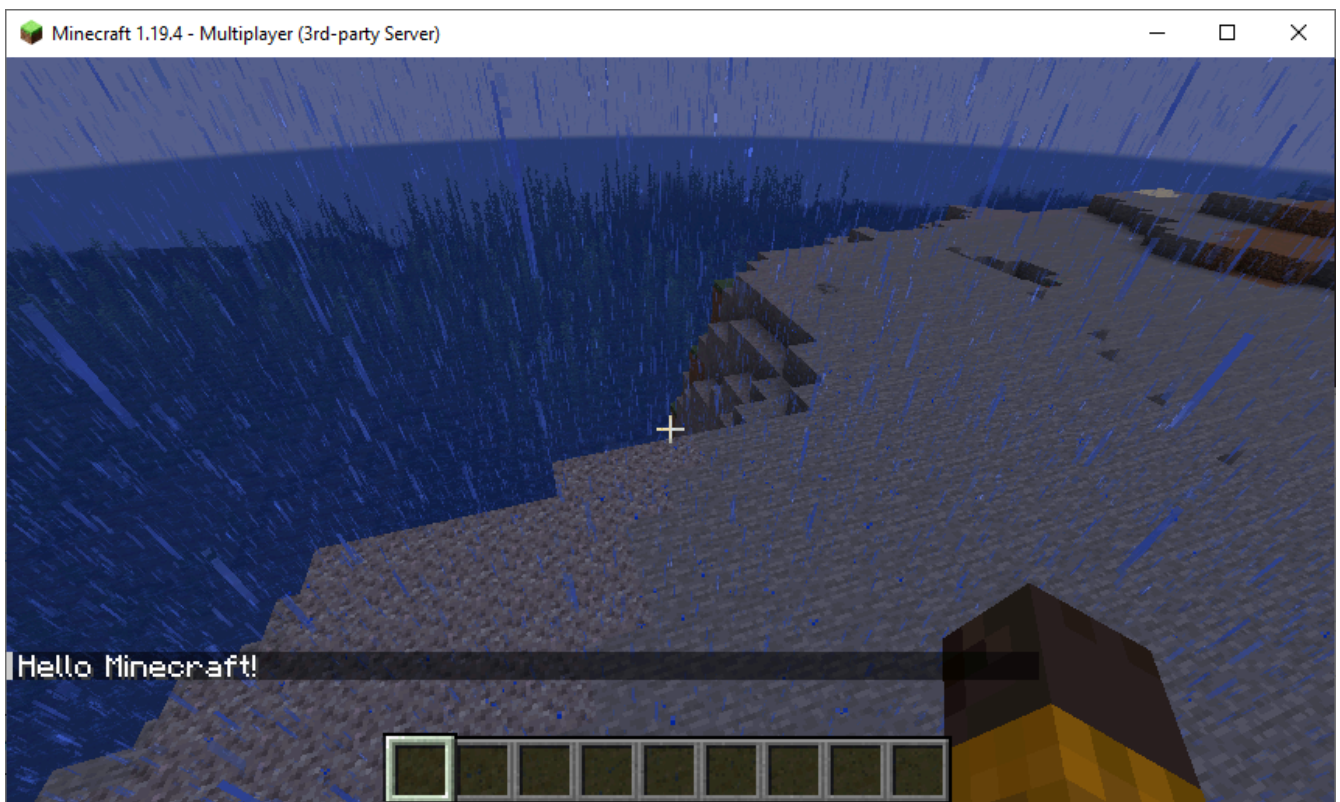
Note #2: Linux/WSL: If you receive an error message similar to: *'error while loading shared libraries: libmcpp.so: cannot open shared object file: No such file or directory'*, run ``sudo ldconfig`` to add **mcpp** to your path.

Note #3: While most compiler flags can be added in any order, `-lmcpp` must be at the end of the command for it to work.

Make sure to have the Server running, then run the program:

```
./hello_minecraft
```

You should see "Hello Minecraft!" pop up in the terminal where your Server is running. If the game is running and connected to the Server, you should also see the message appear in chat:



Common Issues / FAQs

- When you tab away from Minecraft to run a C++ or LC-3 program, the game will pause, which can be mildly annoying. To disable this, hold the F3 key, then press P.
- If you would like to be able to access the WSL filesystem through Windows File Explorer, enable Network Discovery, and you should see "Linux" pop up in the sidebar on the left.