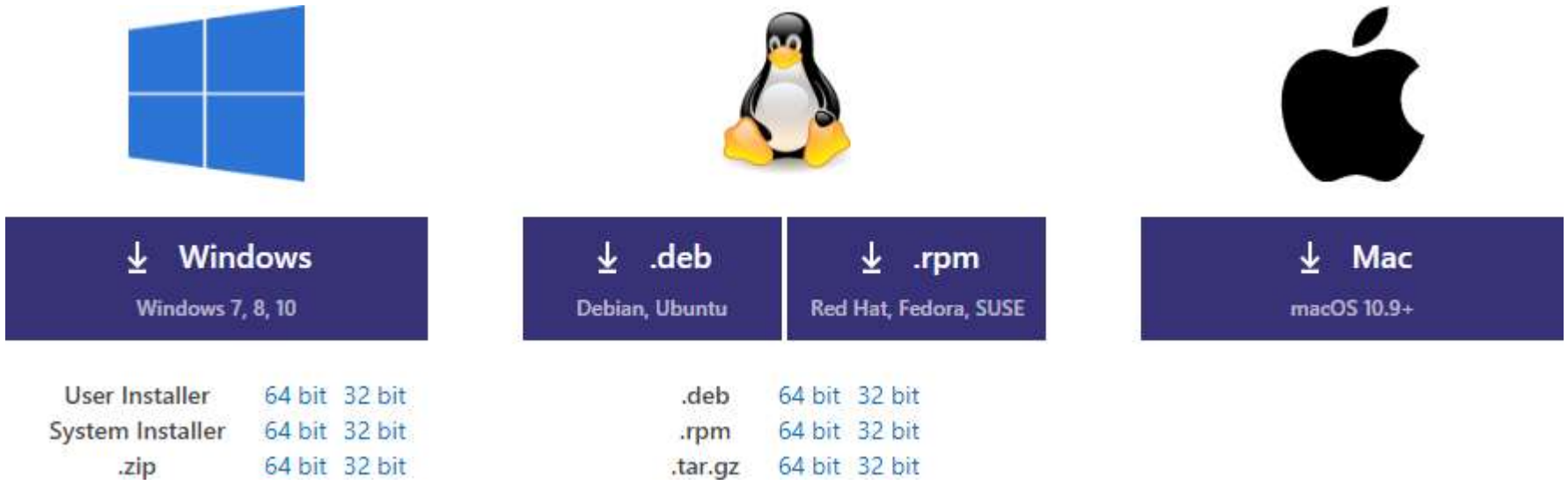
Lab 1

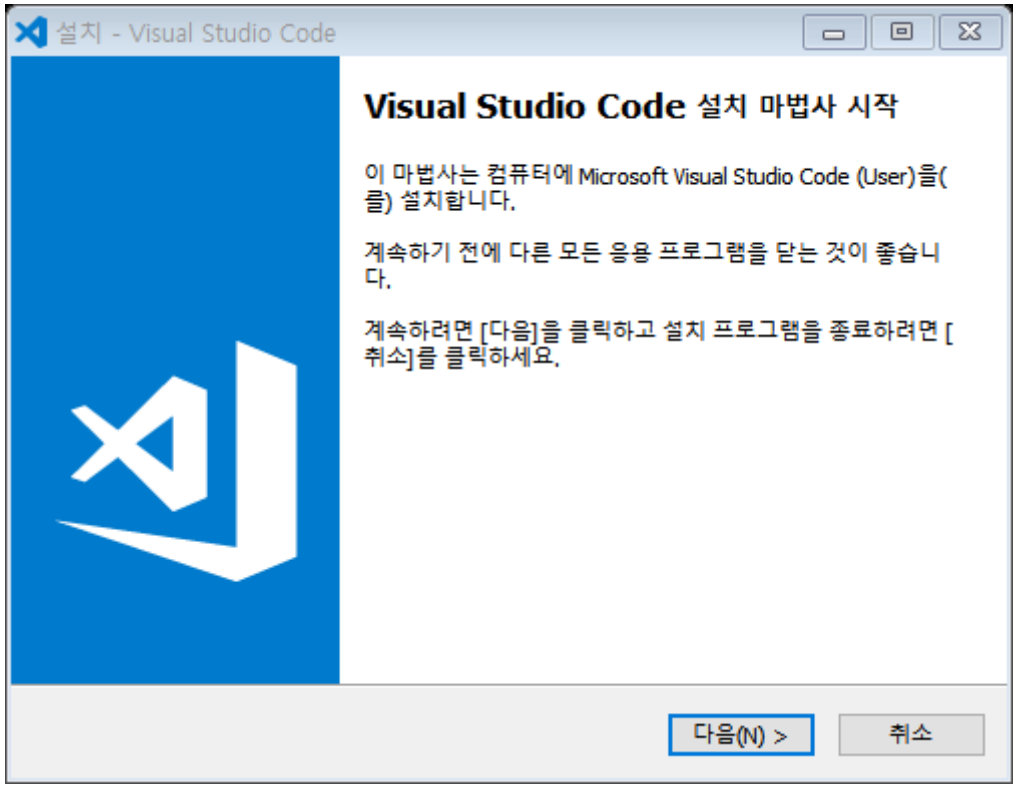
Introduction of Programming Environment

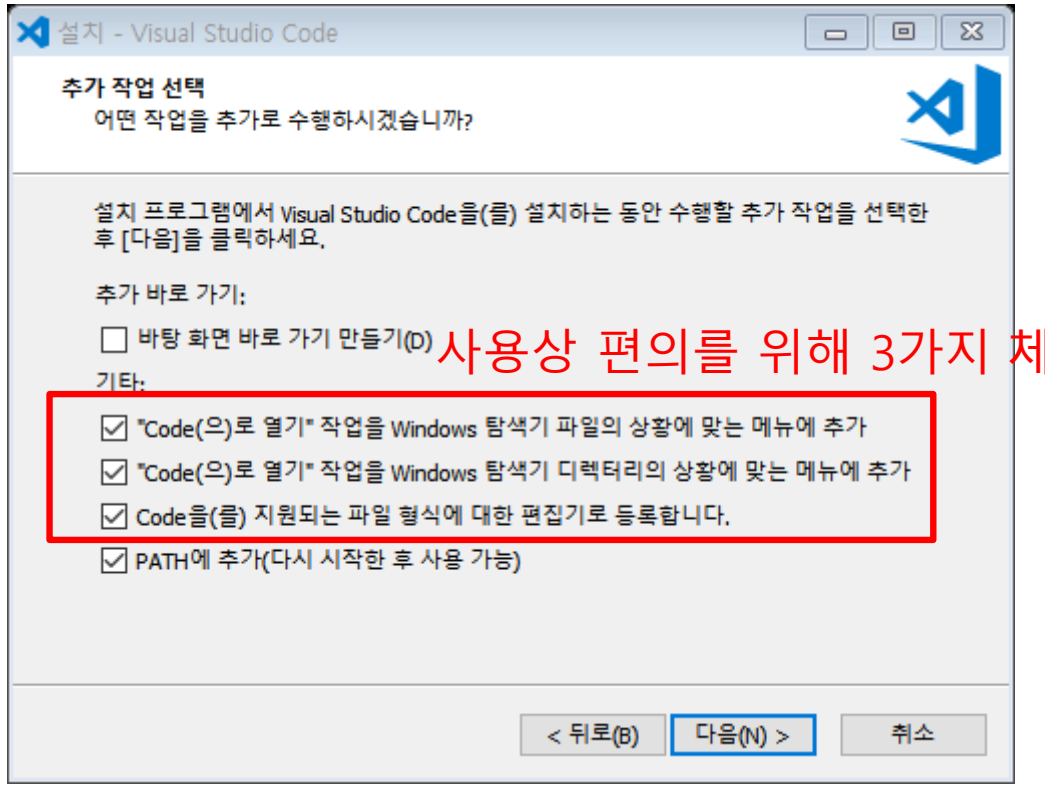
C++ Programming Environment Setup

- Visual Studio Code for C++ Programming
- **https://code.visualstudio.com/download**

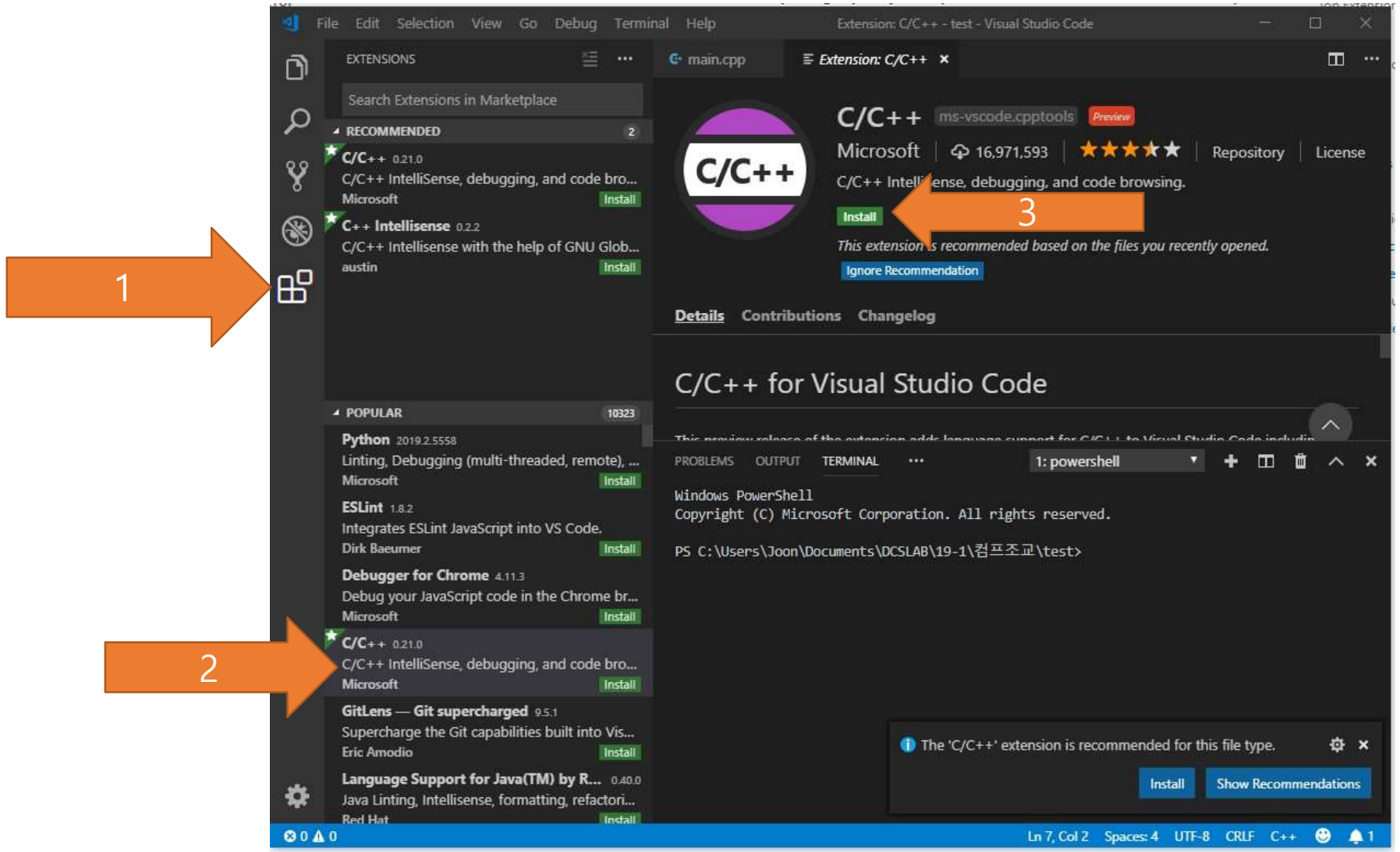


C++ Programming Environment Setup



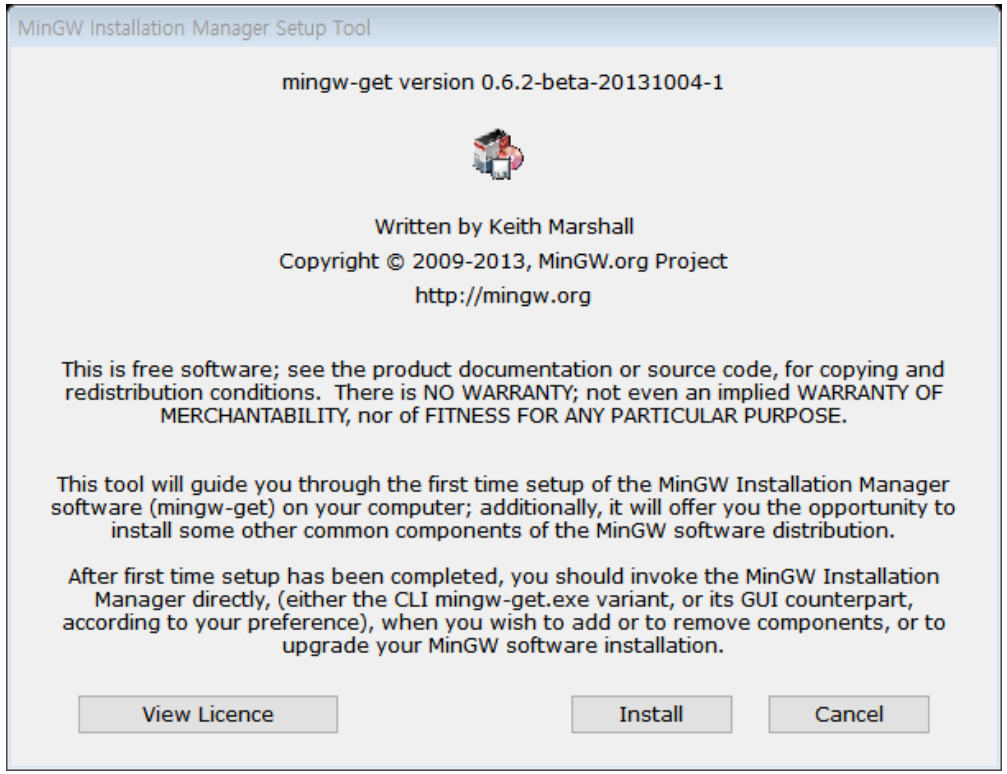


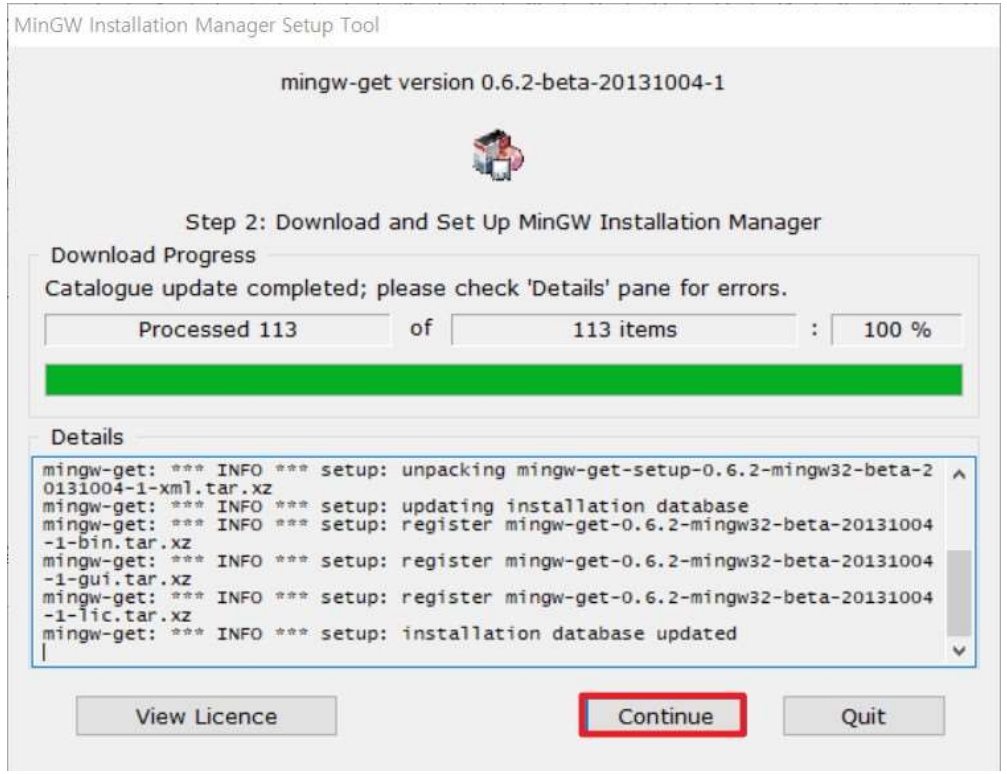
사용상 편의를 위해 3가지 체크


1
2
3
File Edit Selection View Go Debug Terminal Help
Extension: C/C++ - test - Visual Studio Code
EXTENSIONS
Search Extensions in Marketplace
RECOMMENDED 2
C/C++ 0.21.0
C/C++ IntelliSense, debugging, and code bro...
Microsoft Install
C++ Intellisense 0.2.2
C/C++ Intellisense with the help of GNU Glob...
austin Install
POPULAR 10323
Python 2019.2.5558
Linting, Debugging (multi-threaded, remote), ...
Microsoft Install
ESLint 1.8.2
Integrates ESLint JavaScript into VS Code.
Dirk Baeumer Install
Debugger for Chrome 4.11.3
Debug your JavaScript code in the Chrome br...
Microsoft Install
C/C++ 0.21.0
C/C++ IntelliSense, debugging, and code bro...
Microsoft Install
GitLens — Git supercharged 9.5.1
Supercharge the Git capabilities built into Vis...
Eric Amodio Install
Language Support for Java(TM) by R... 0.40.0
Java Linting, Intellisense, formatting, refactori...
Red Hat Install
main.cpp
Extension: C/C++
C/C++
ms-vscode.cpptools
Preview
Microsoft
16,971,593
Repository
License
C/C++ Intellisense, debugging, and code browsing.
Install
This extension is recommended based on the files you recently opened.
Ignore Recommendation
Details Contributions Changelog
C/C++ for Visual Studio Code
PROBLEMS OUTPUT TERMINAL
1: powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
PS C:\Users\Joon\Documents\DCSLAB\19-1\컴프조교\test>
The 'C/C++' extension is recommended for this file type.
Install
Show Recommendations
0 0
Ln 7, Col 2 Spaces: 4 UTF-8 CRLF C++ 1

C++ Programming Environment Setup

- For gcc compile, setup mingw
- https://sourceforge.net/projects/mingw/





C++ Programming Environment

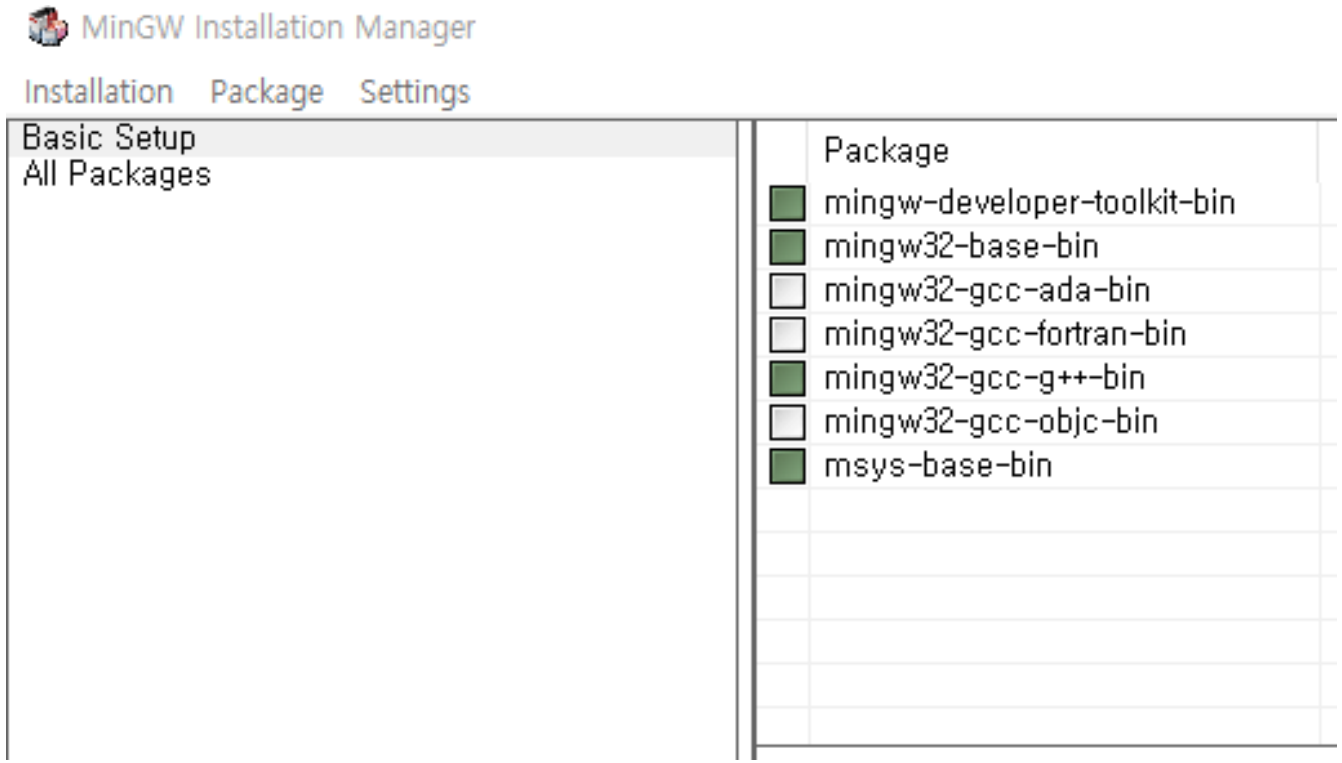
- Check package

mingw-developer-toolkit,

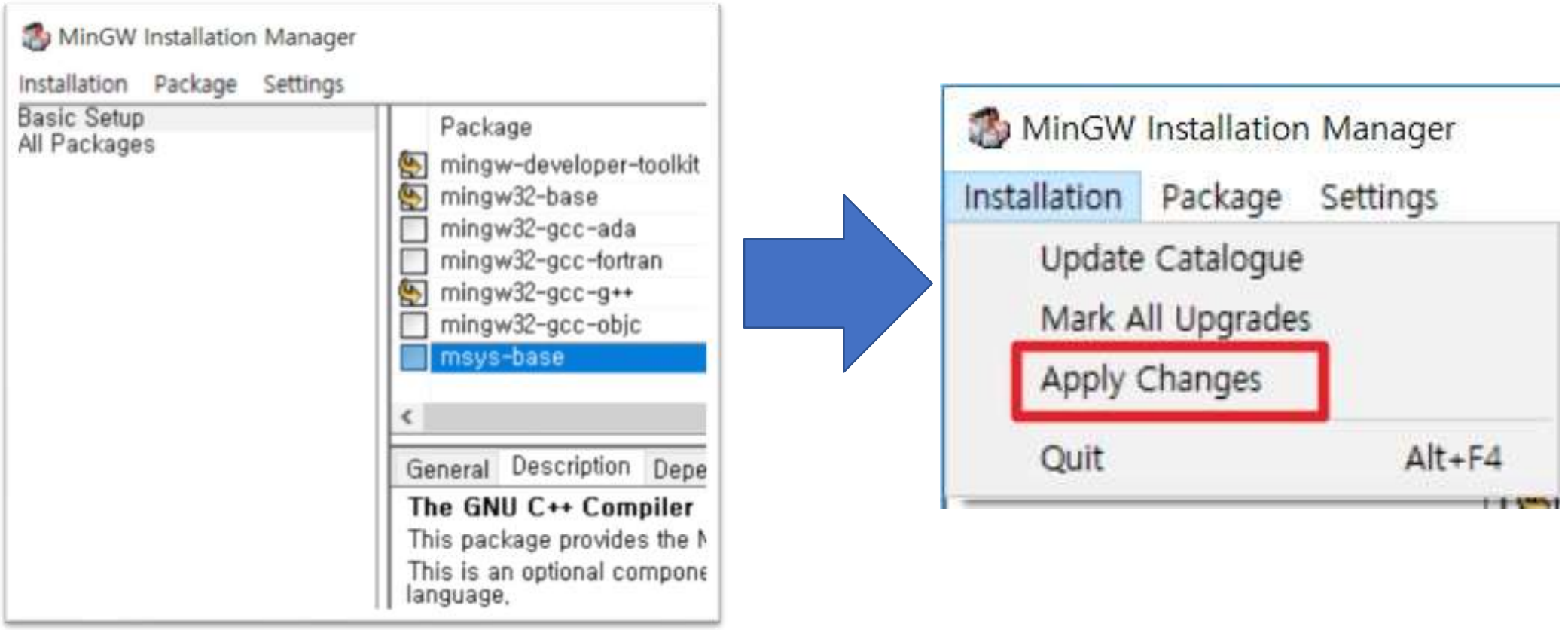
mingw32-base,

mingw32-gcc-g++,

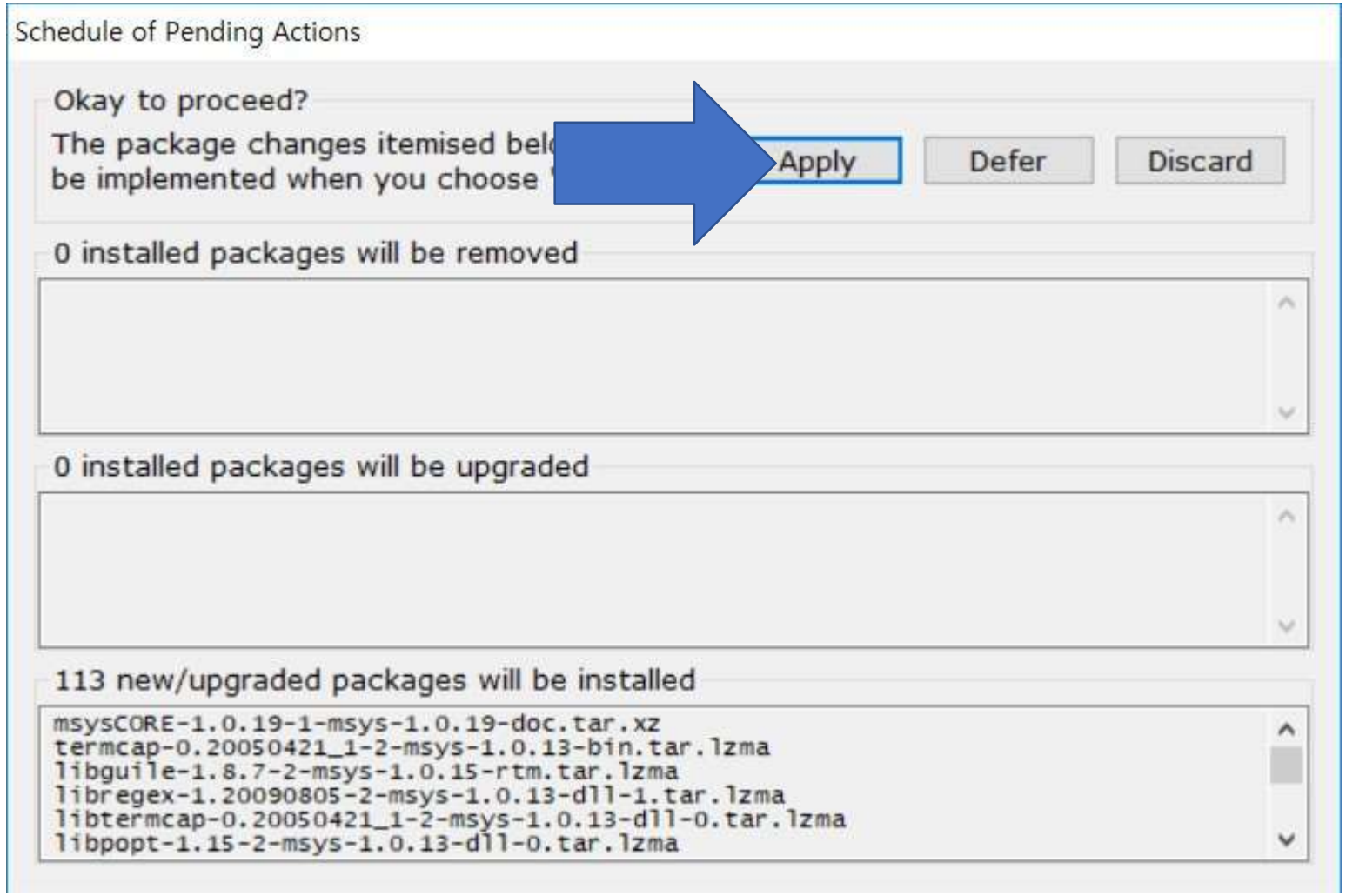
msys-base-bin



C++ Programming Environment

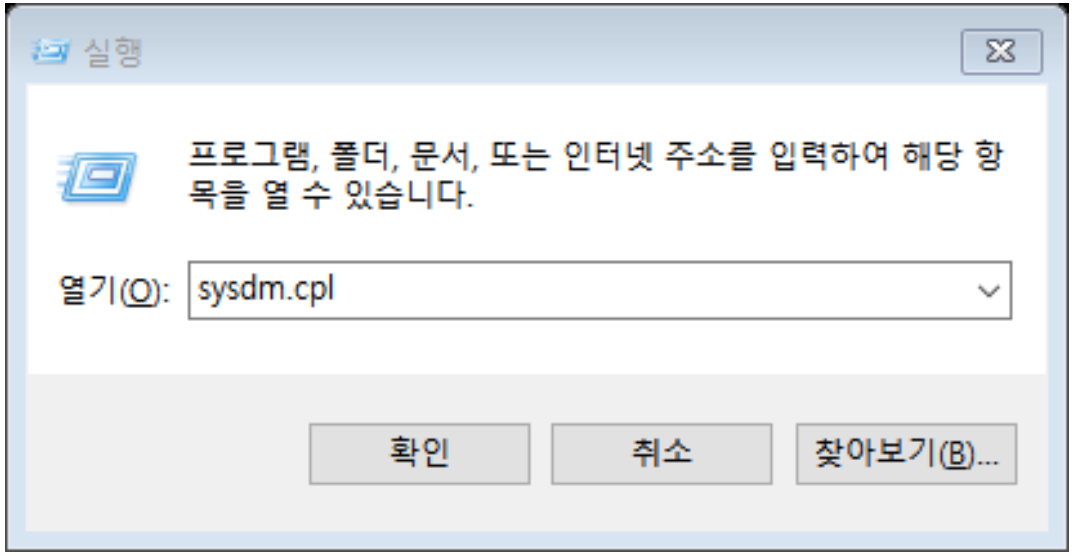


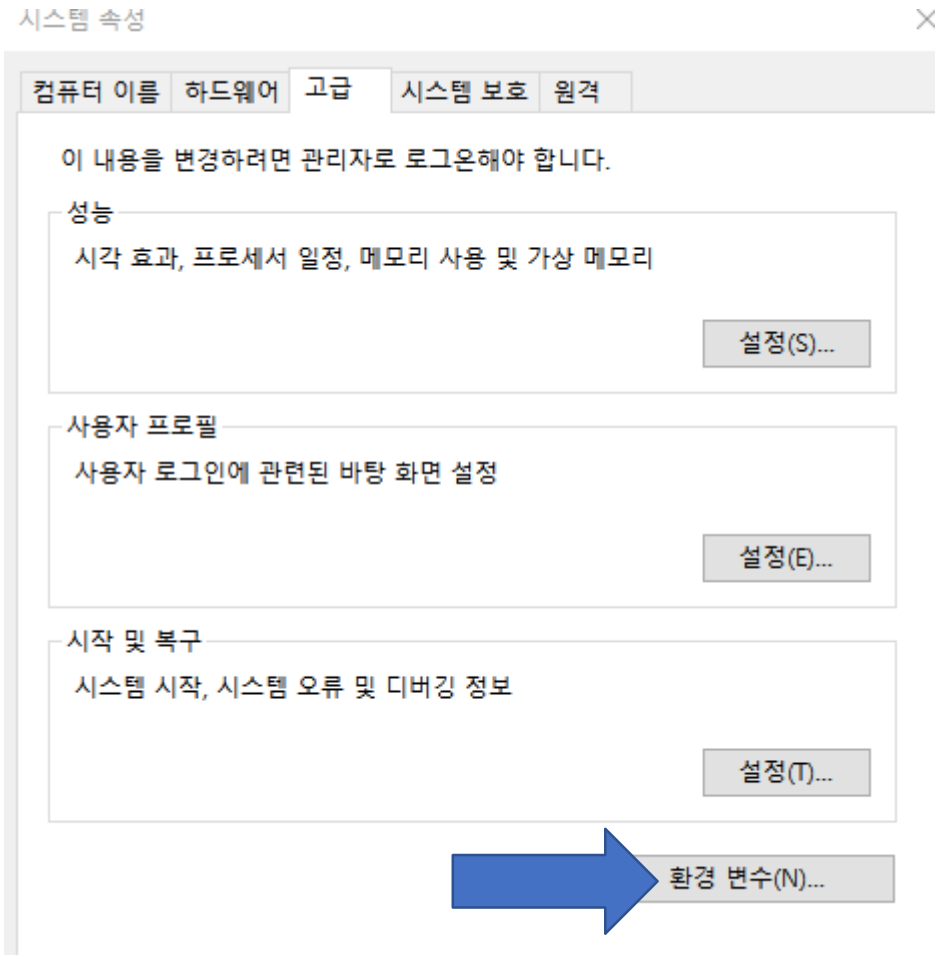
C++ Programming Environment



C++ Programming Environment

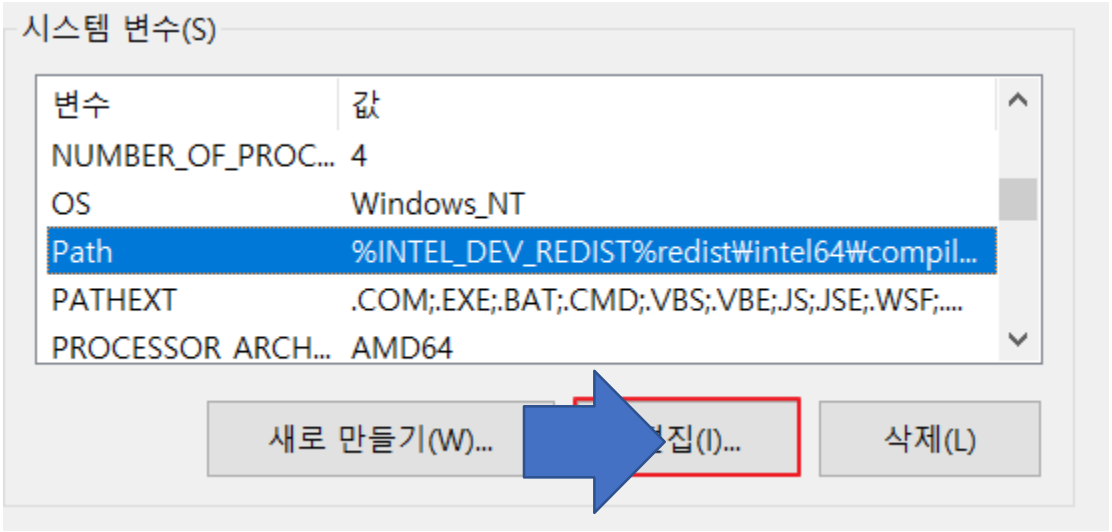
- Edit system path
- Press the window key + R, then type
- sysdm.cpl to run system properties in Control Panel.



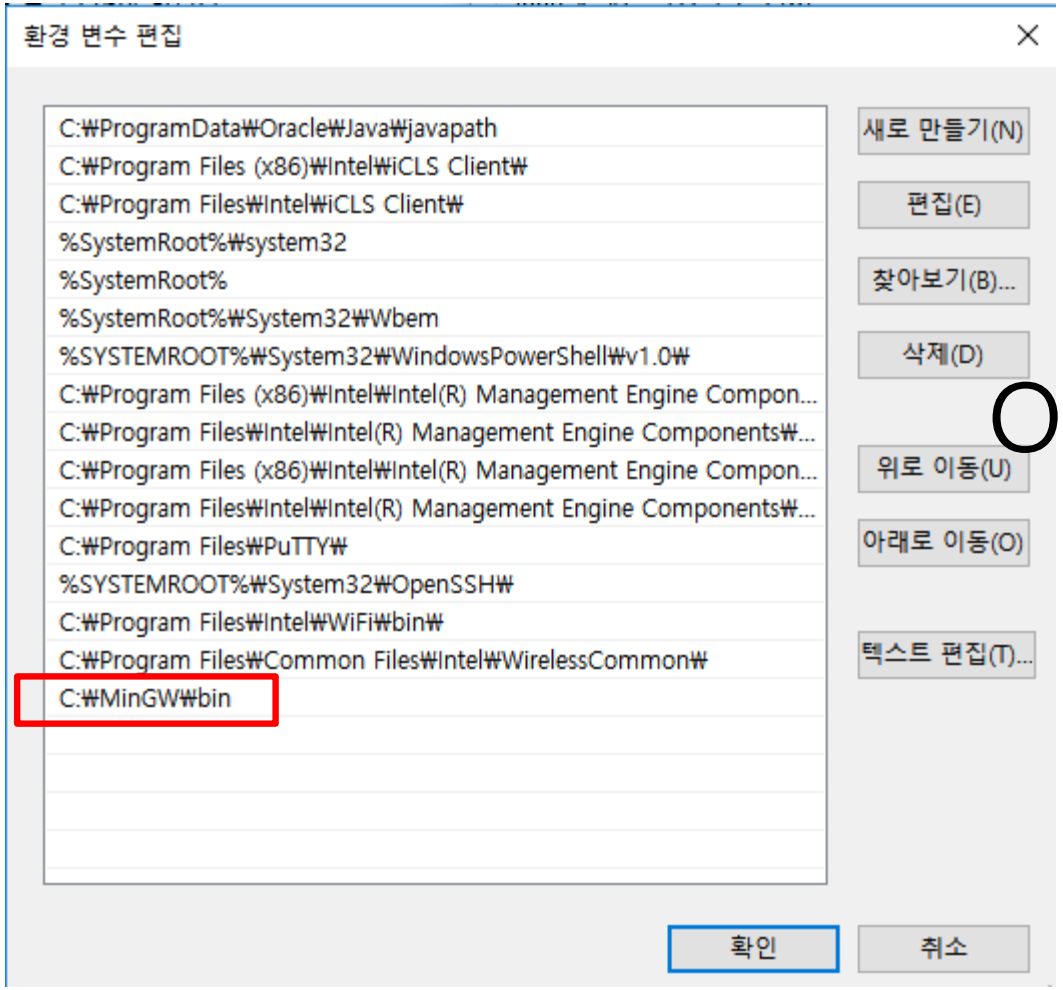


C++ Programming Environment

- Under System Variables, select Path and click the Edit button.

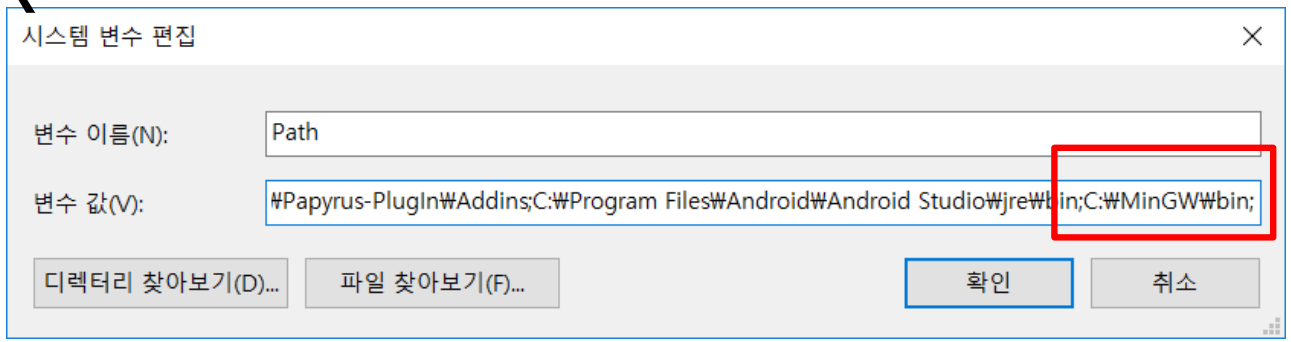


C++ Programming Environment



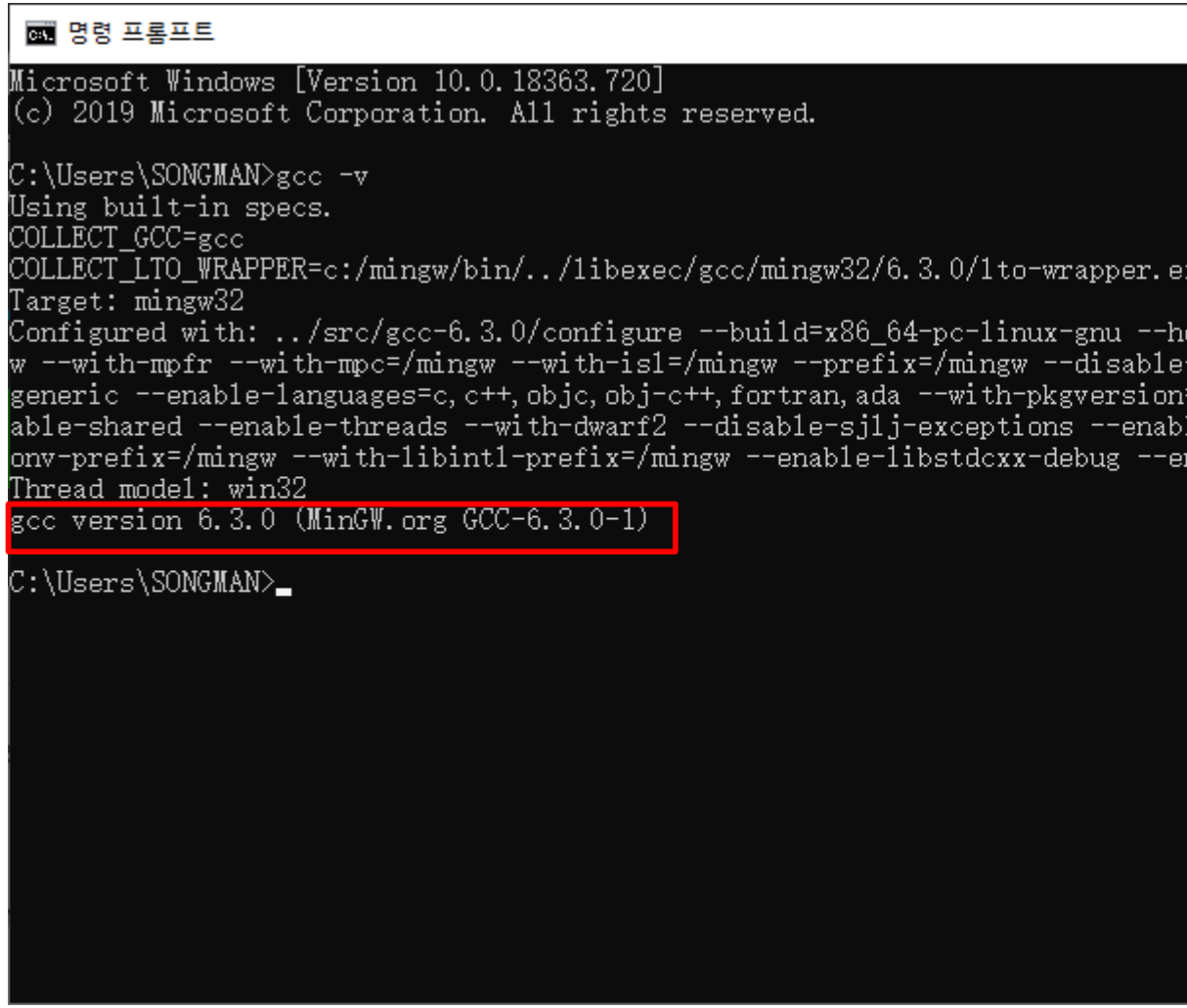
Depending on the version of the window, click New to add the path C:₩MinGW₩bin or add ;C:₩MinGW₩bin to the end of the path variable value.

OR



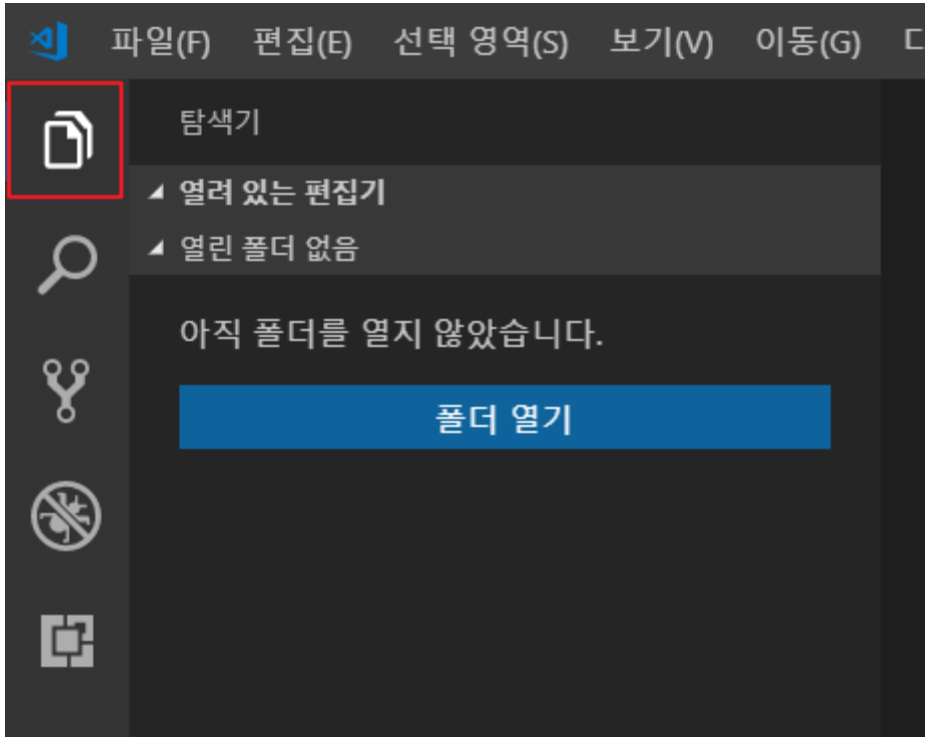
C++ Programming Environment

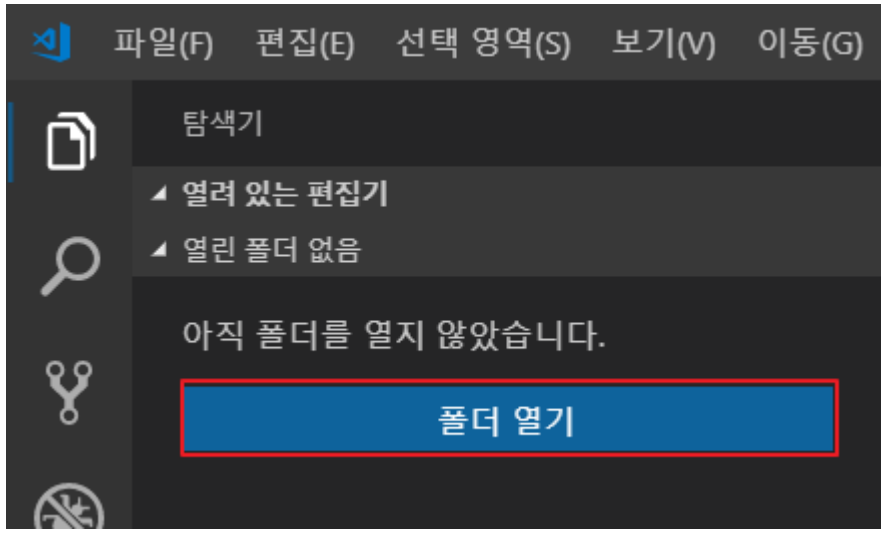
- If the progress is successful, you can check the gcc, g++ version information at the command prompt as follows:



C++ Programming Environment

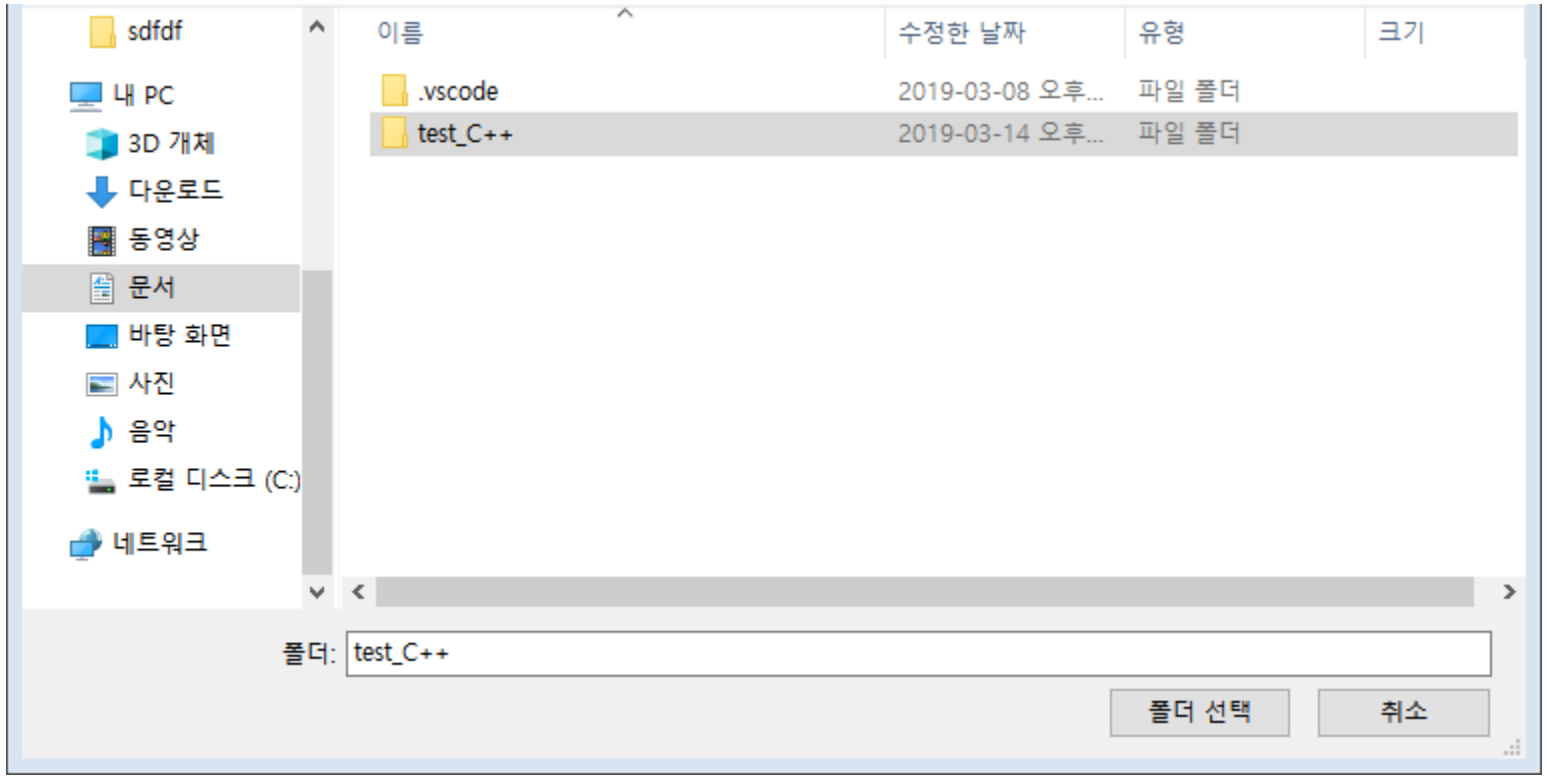
- Click the Explorer icon in the activity bar located on the left, or press the shortcut Ctrl + Shift + E to open the Explorer on the sidebar as shown in the capture screen below.





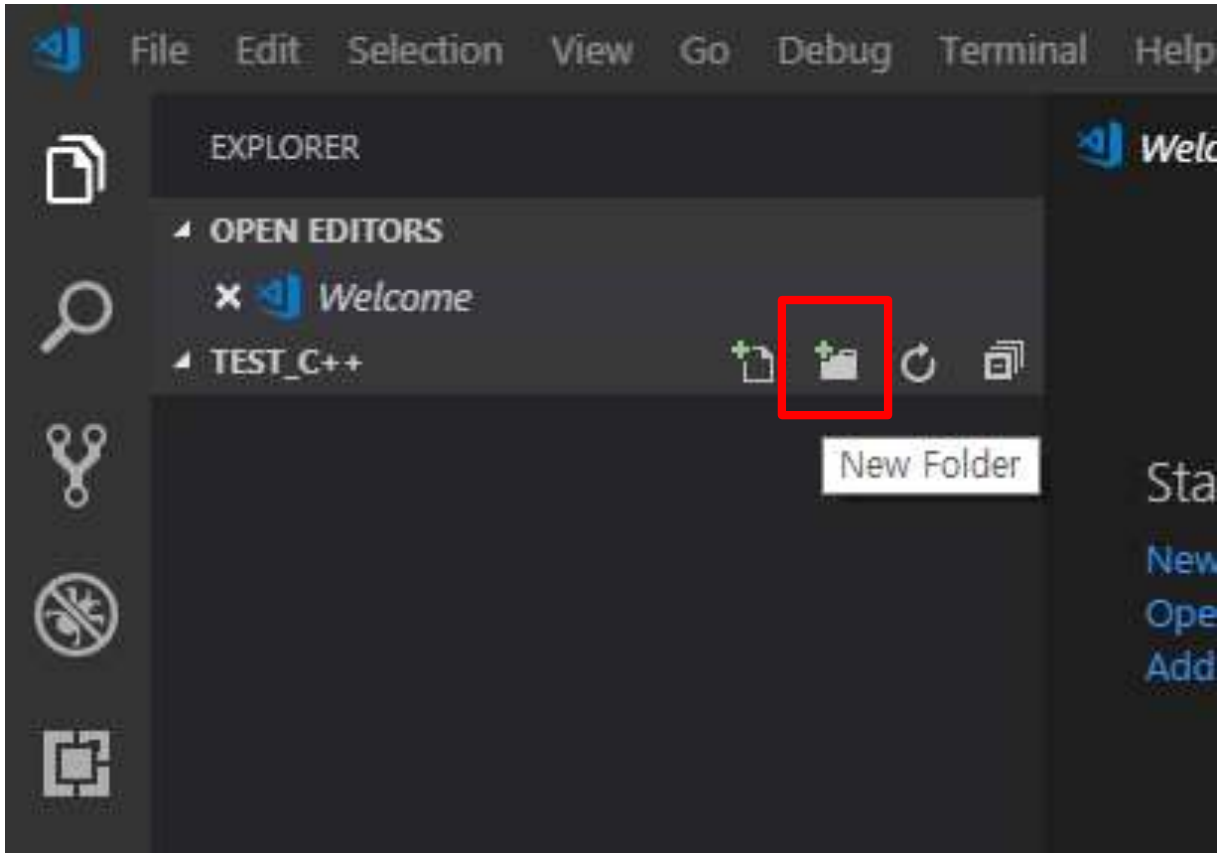
C++ Programming Environment

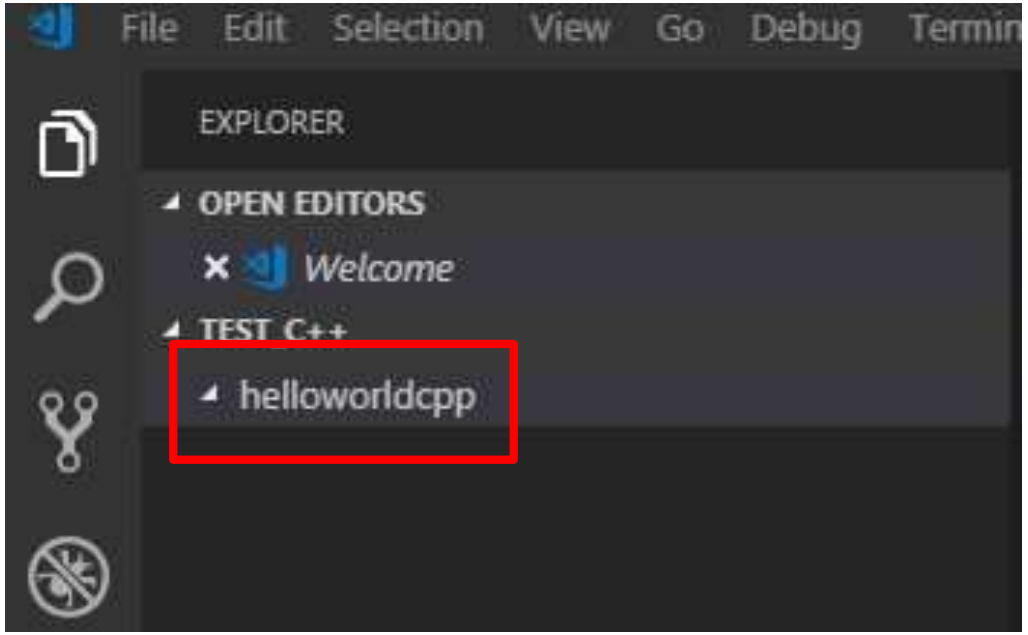
- Create the test_C++ folder and click the Select Folder button.



C++ Programming Environment

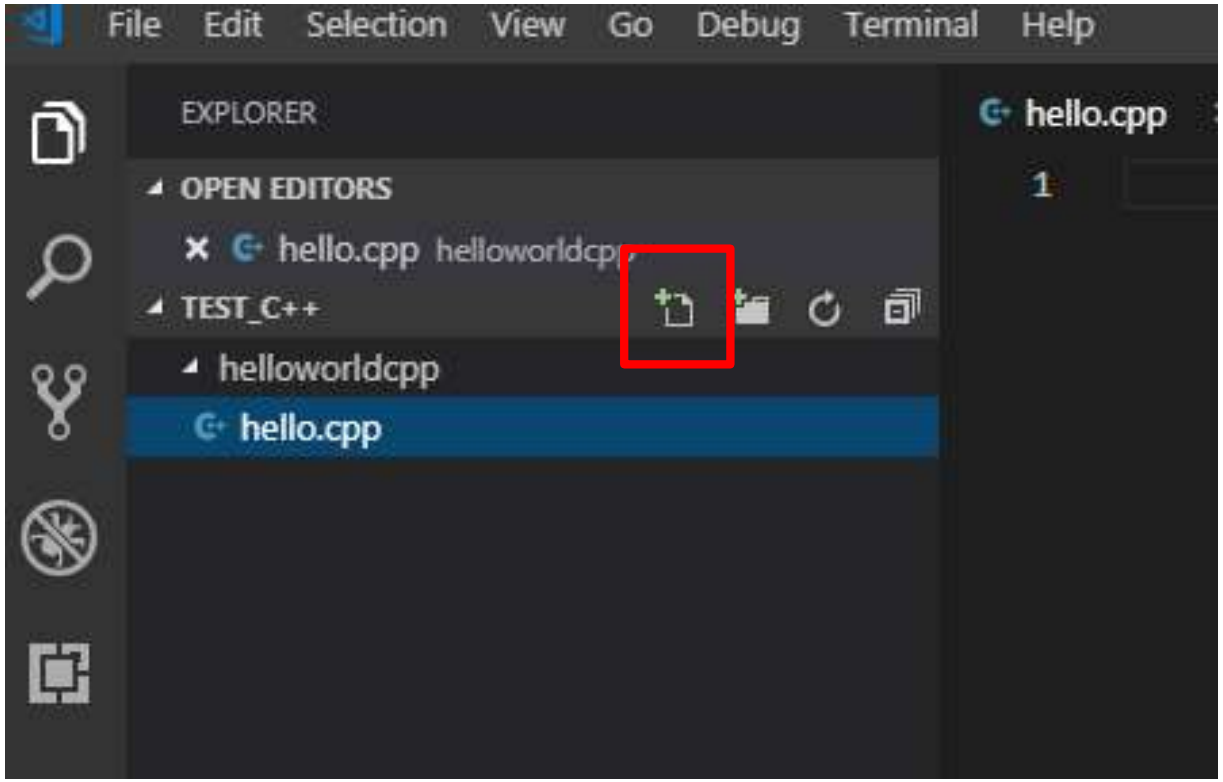
- Click the new folder icon and create helloworldcpp folder.





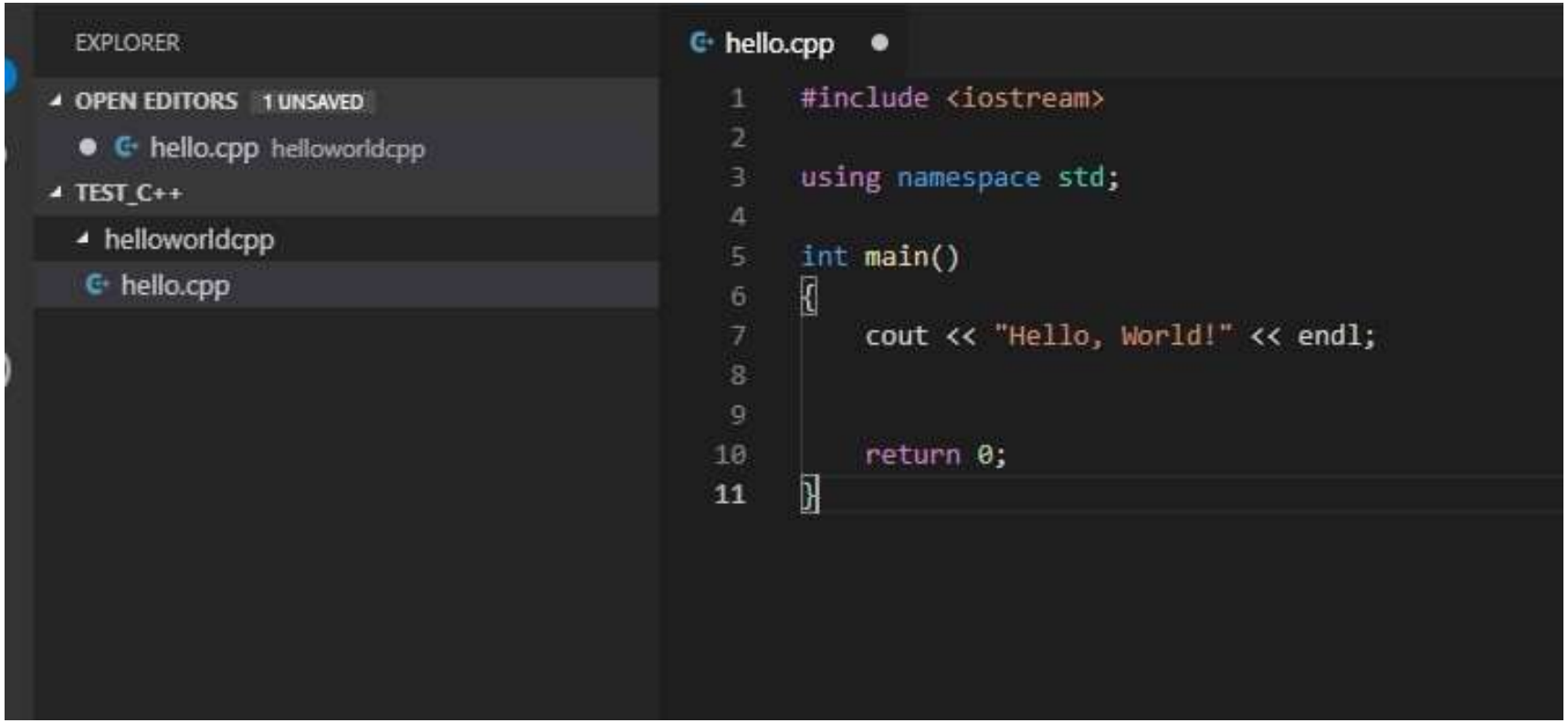
C++ Programming Environment

- Click the new file icon and create hello.cpp



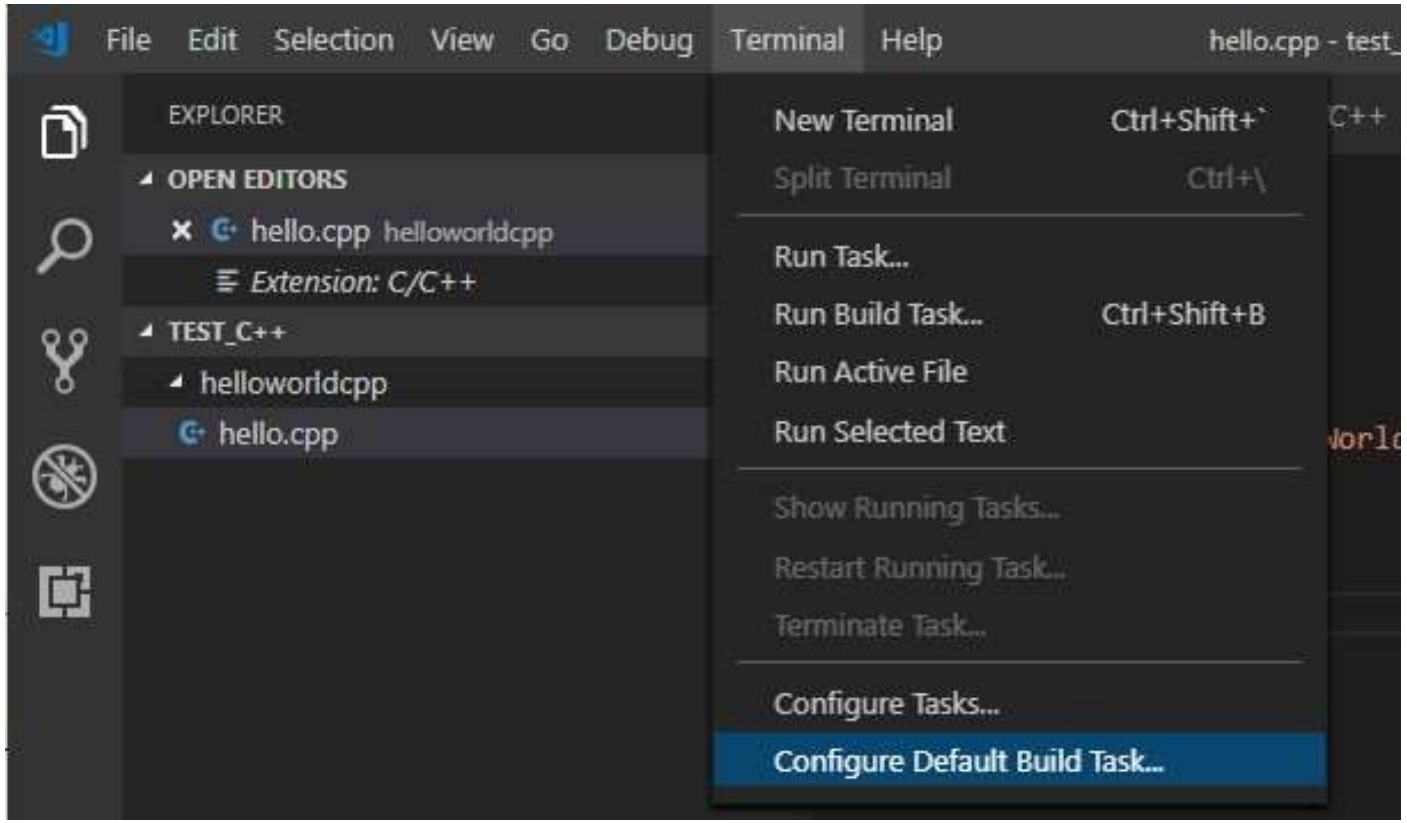
C++ Programming Environment

- Enter the following code into the Hello.cpp file and press Ctrl + S to save.



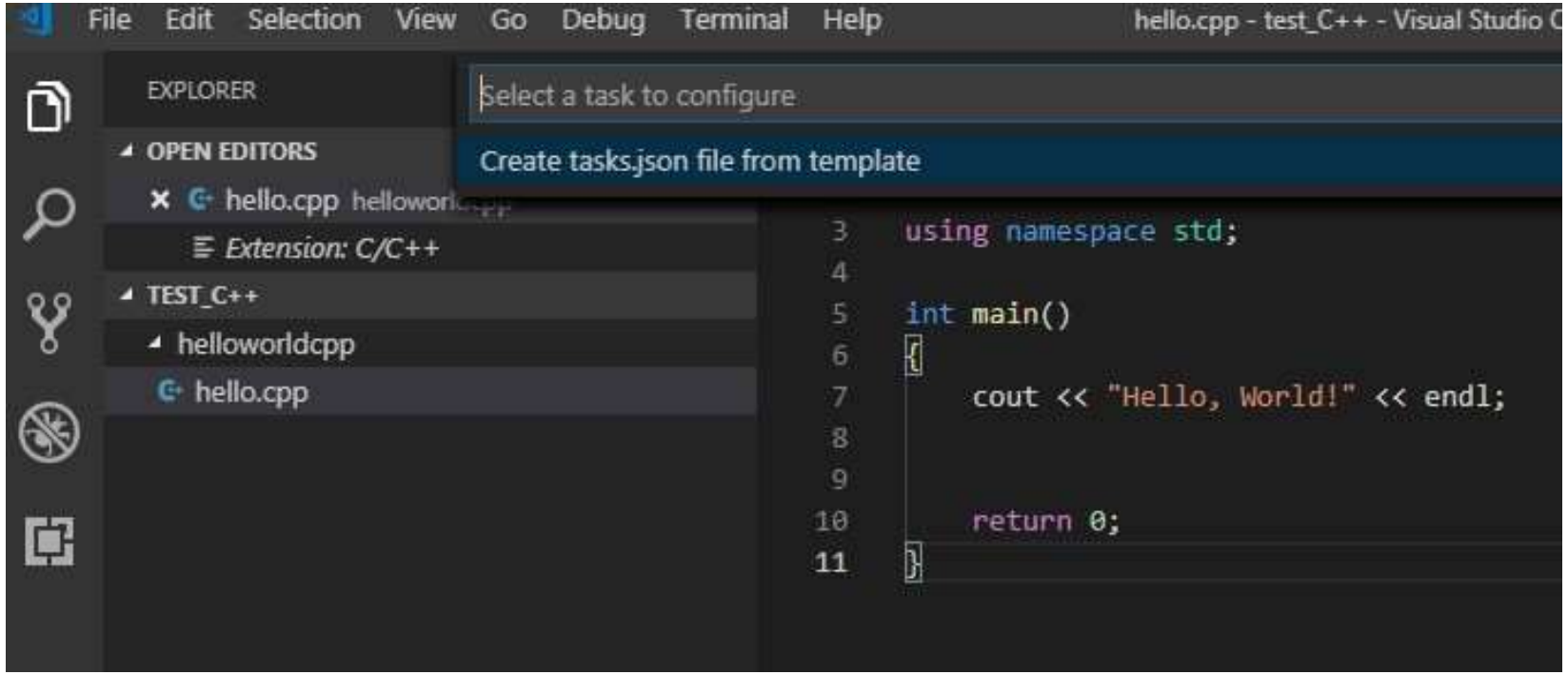
C++ Programming Environment

- From the menu of the Visual Studio Code, select Terminal > Default Build Job Configuration.



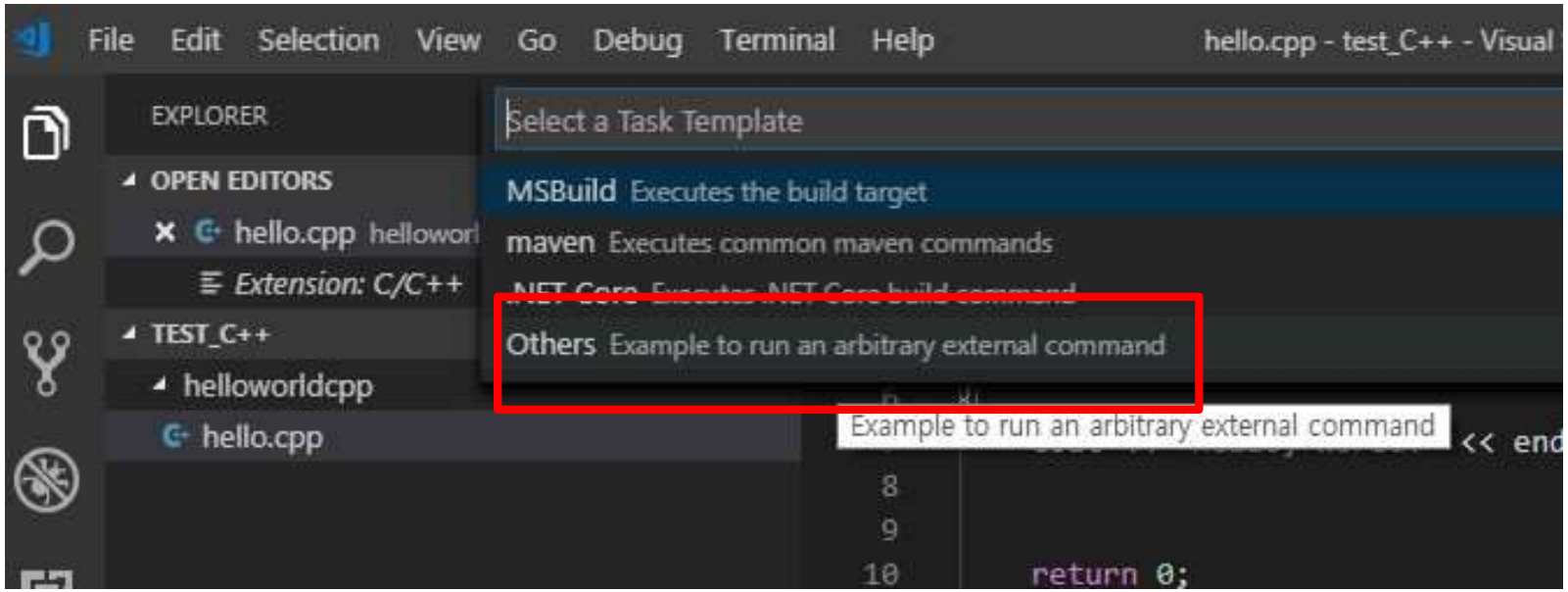
C++ Programming Environment

- Click Create tasks.json file from template.



C++ Programming Environment

- Click Others.



C++ Programming Environment

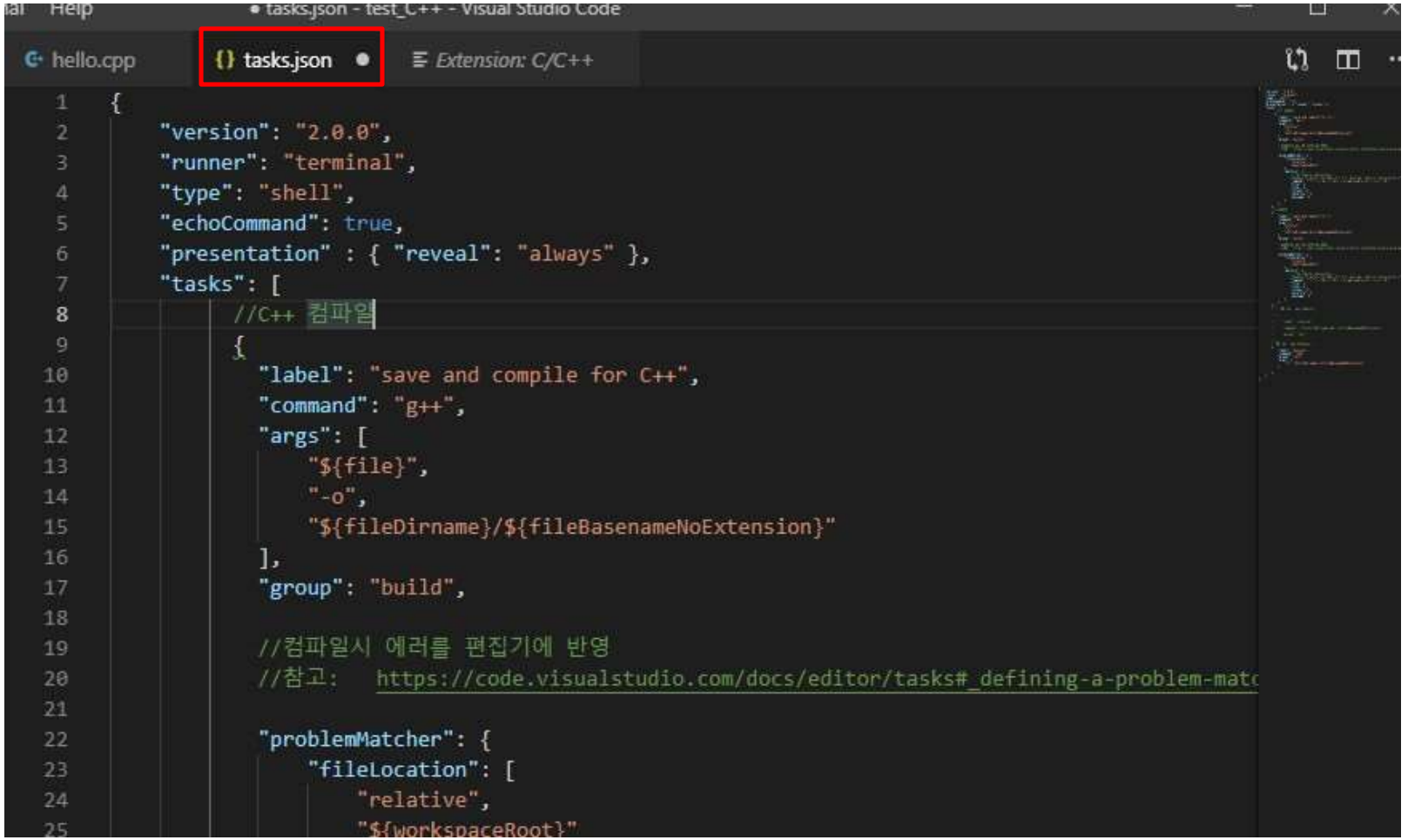
- Copy & Paste code

- **Windows ver.**

https://drive.google.com/file/d/1hTYHum9-3xIv1RKIdGUX8nnA109ohuL-/view?usp=sharing

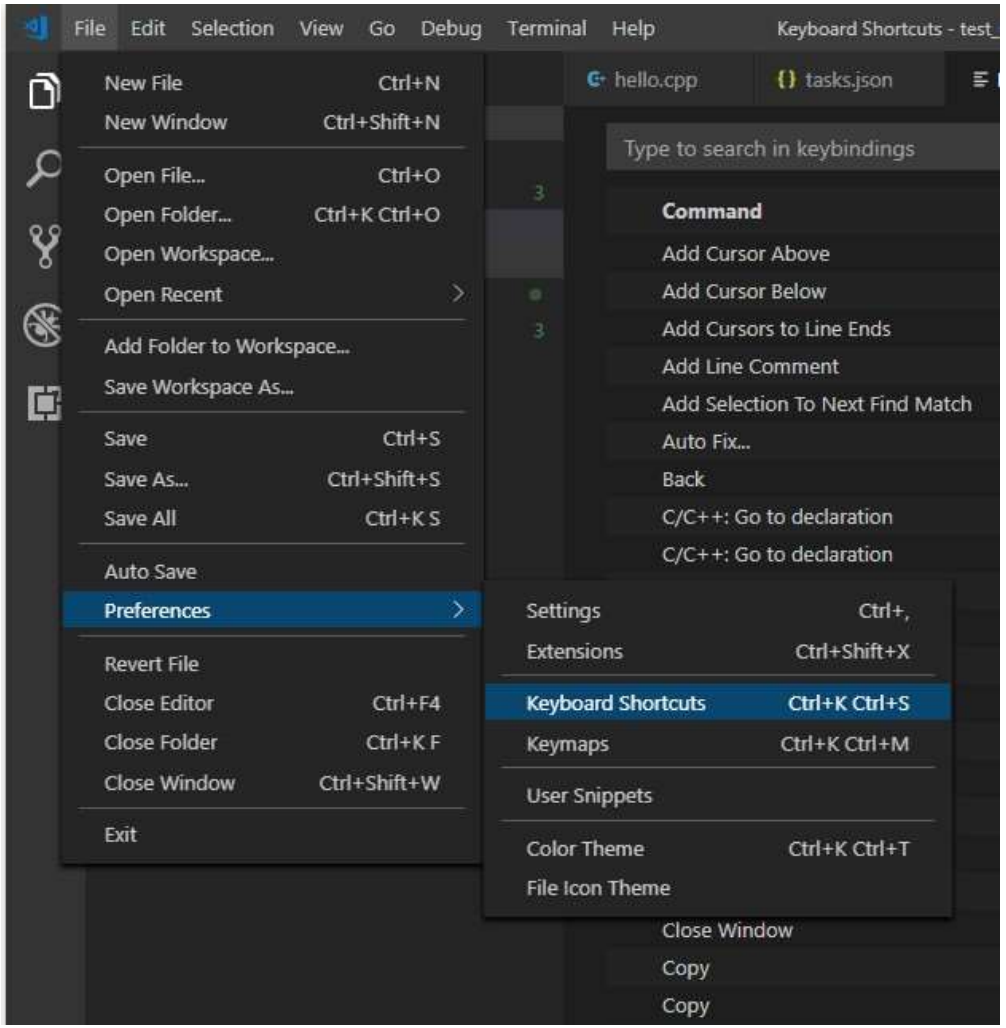
- **Mac ver.**

https://drive.google.com/file/d/1Po_JFwm-k3tMw06faT-UUPILqbhzxx8Z/view?usp=sharing

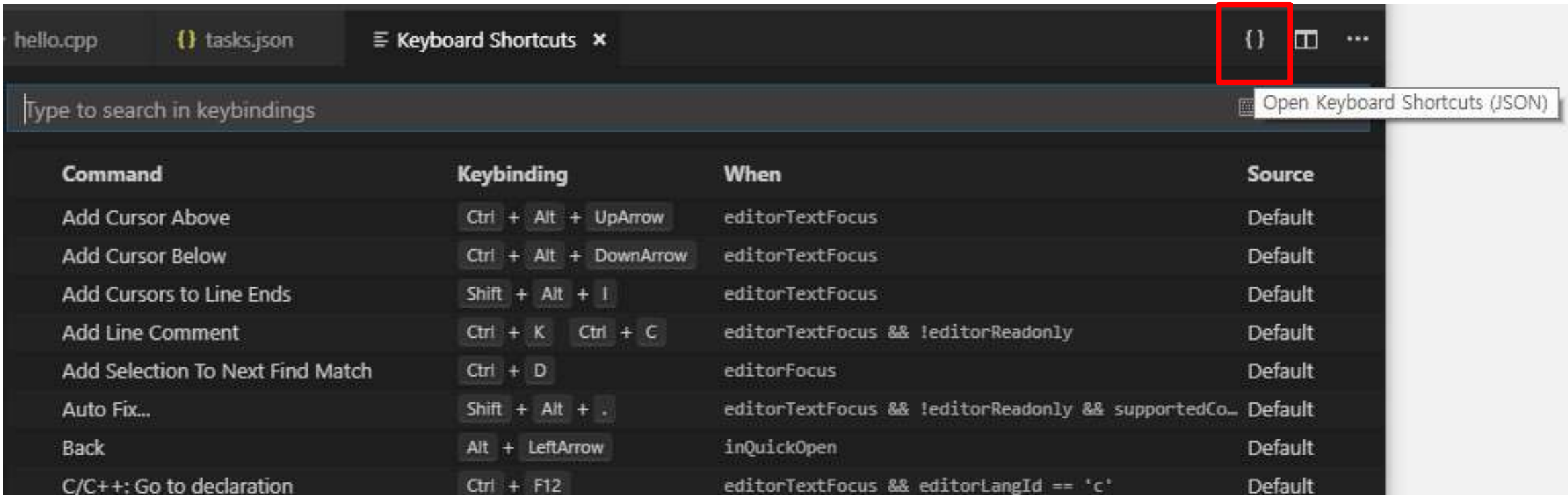


C++ Programming Environment

•For your convenience, set the shortcut key.

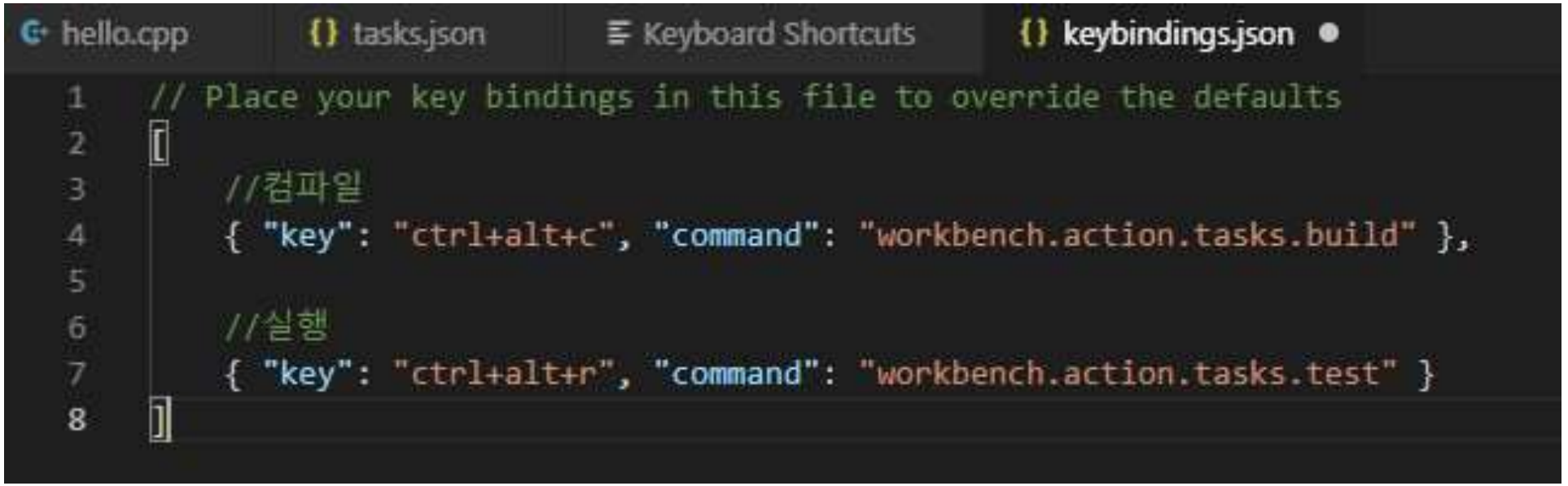


C++ Programming Environment



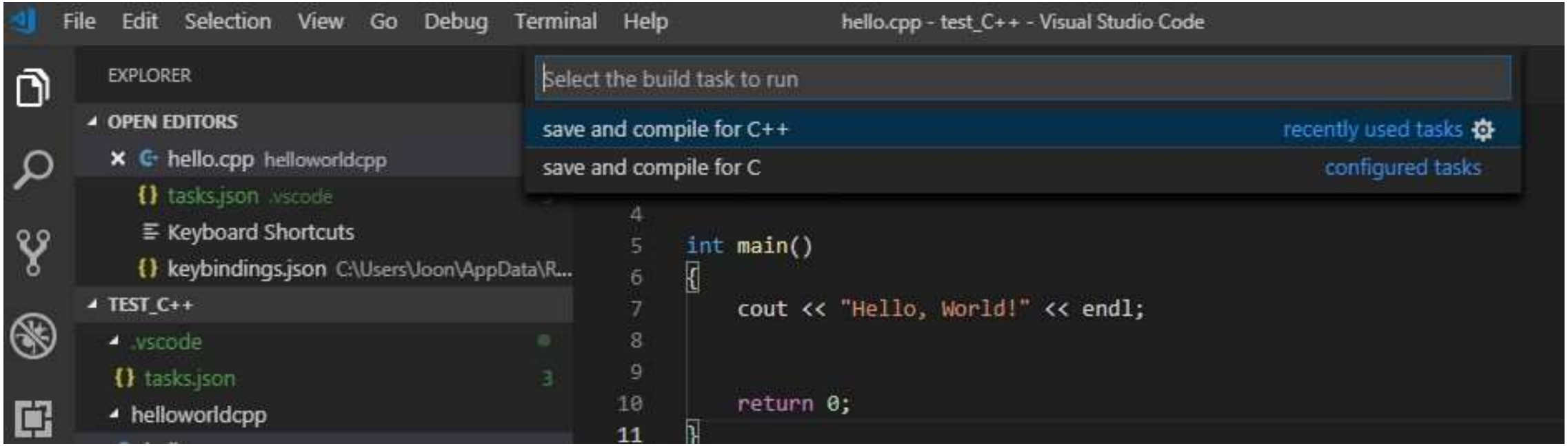
C++ Programming Environment

- Enter and press Ctrl + S to save as follows

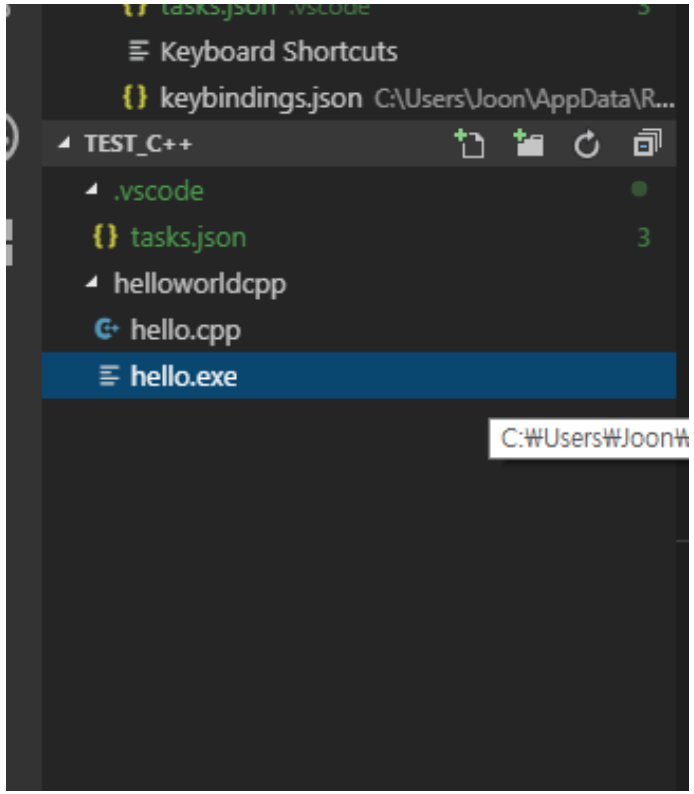


C++ Programming Environment

- In Hello.cpp, press Ctrl +Alt + C and click *save and compile for C++.*



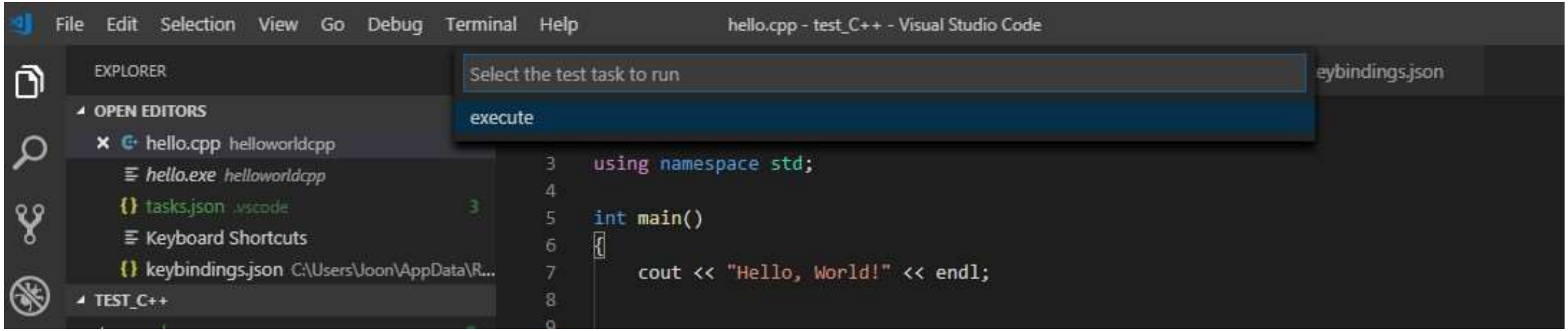
C++ Programming Environment



- All files being edited will be saved and the compilation progress will be shown in the terminal before.
- If the compile was run without any problems, the file Hello.exe, the compilation result, will be displayed in the left navigator.

C++ Programming Environment

- Press Ctrl + Alt + R and click *execute*
- The results of the execution are displayed in the terminal.




```
Hello, World!

Terminal will be reused by tasks, press any key to close it.
```

C++ Example code 1

- Try to compile this code and print out the results.
- What functions should add ed to change private variab le? (indirect approach)

```
1   #include <iostream>
2   #include <string>
3   using namespace std;
4
5   class Item { // Class definition
6       public:
7           string title;
8           double price;
9           double SalePrice() { return (price*0.9);}
10          bool isAvailable() { return (inStockQuantity > 0); }
11      private:
12          int inStockQuantity;
13      };
14
15      int main(void)
16      {
17          Item a;
18          a.title="comp";
19          a.price=2000;
20          cout << a.title <<endl;
21          cout << a.SalePrice() << endl;
22          return 0;
23      }
24
```

C++ Example code 2

- Try to compile this code and print out the results.

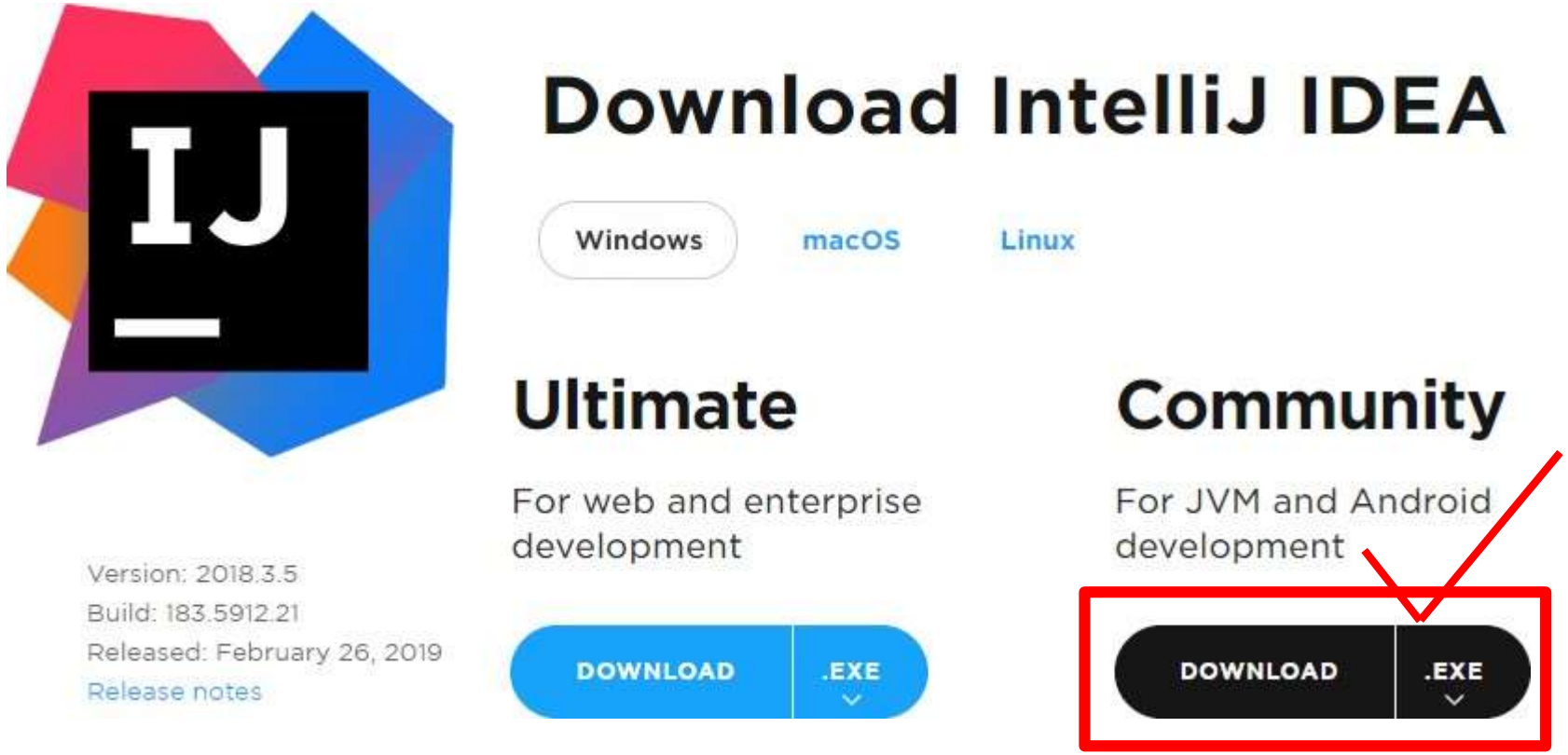
```cpp
#include <iostream>
#include <string>
#include <cstring>
#include <assert.h>
using namespace std;

class String {
    public:
        String(const char *s) {
            len = strlen(s);
            str = new char[len + 1];
            assert(str != 0);
            strcpy(str,s);
        }

        ~String() { delete [] str; }
        void showStr()
        {
        cout<<str<<endl;
        }

    private:
        int len;
        char *str;
};

int main(void)
{
    String str = String("str"); // Definition
    str.showStr();
    return 0;
}
```

Java Programming Environment Setup

- Download Java JDK and set Path
- https://codedragon.tistory.com/8510
- https://limkydev.tistory.com/61
- For reference

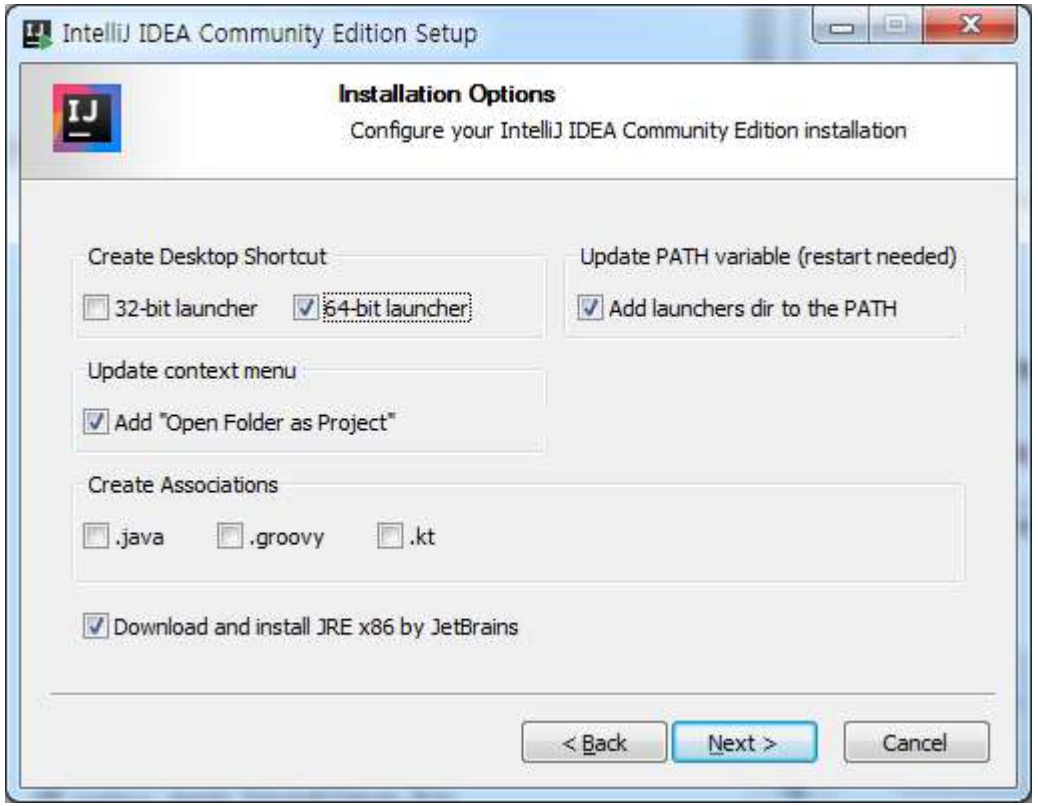
Java Programming Environment Setup

- https://www.jetbrains.com/idea/



Installation 2

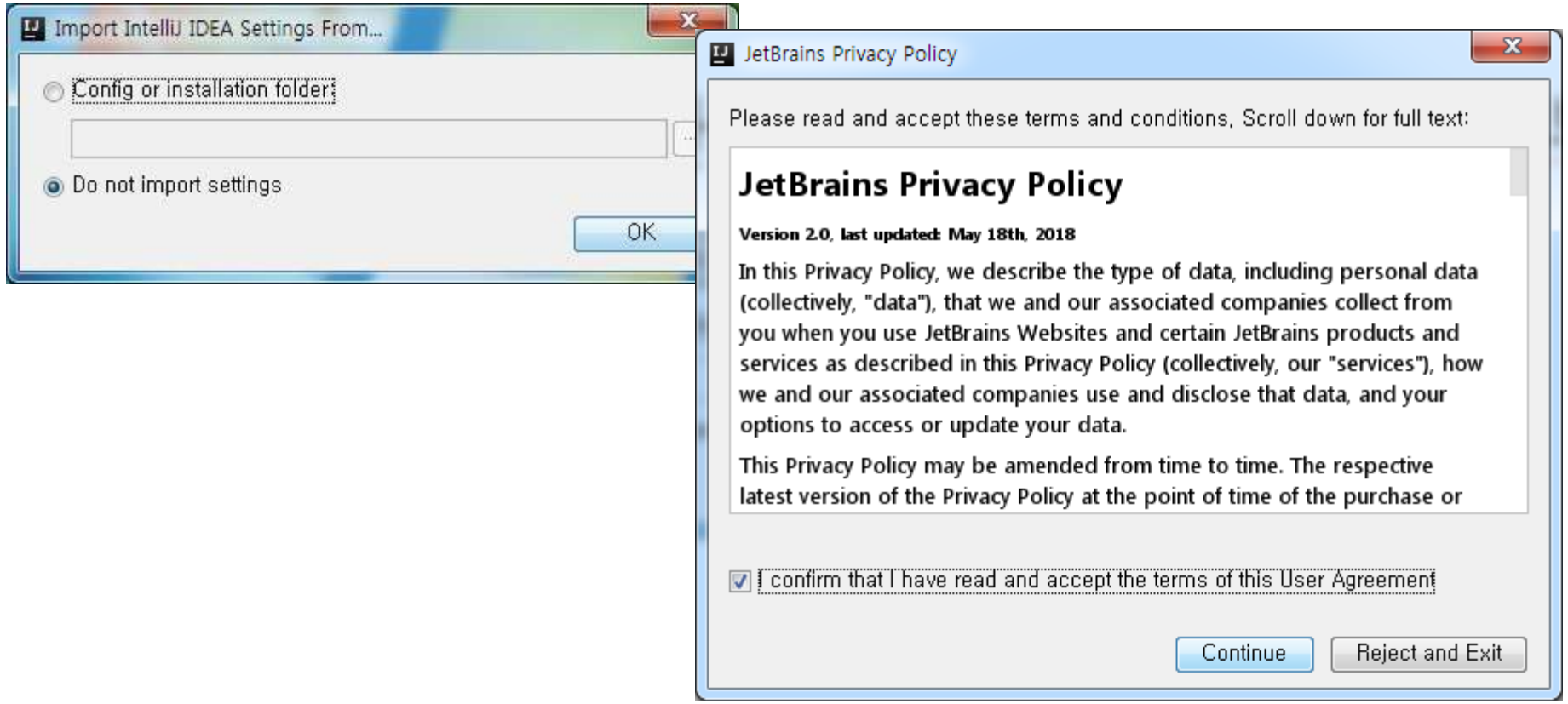
Next > … >



→ Install

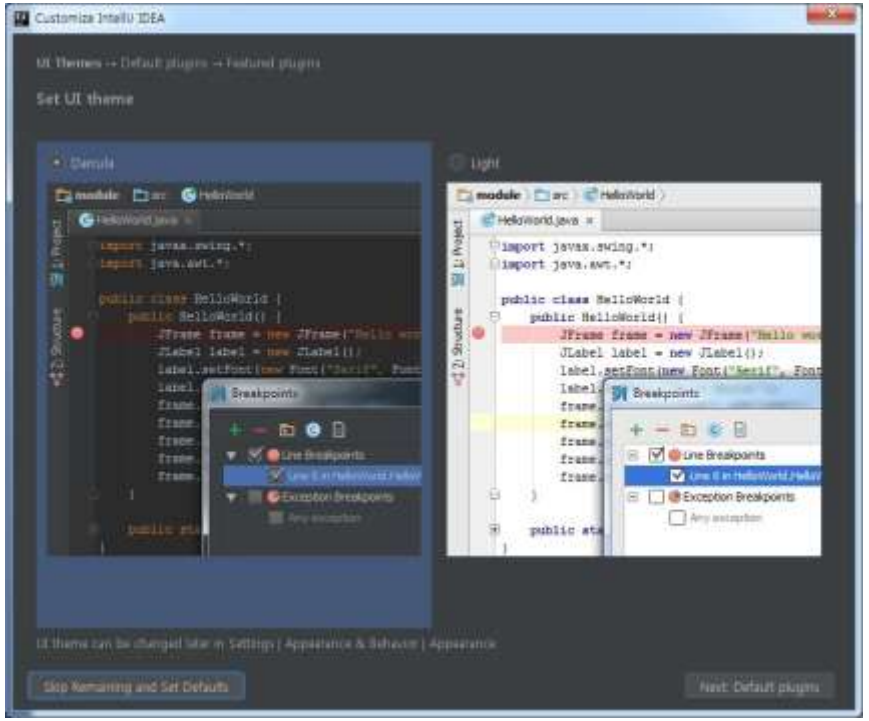
Installation 3

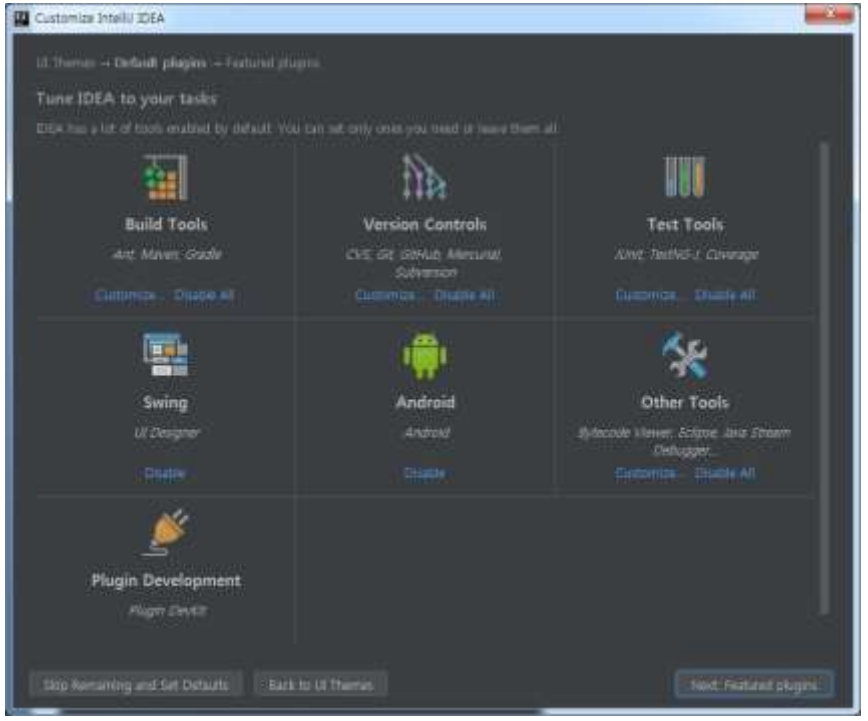
- Reboot the computer and run program



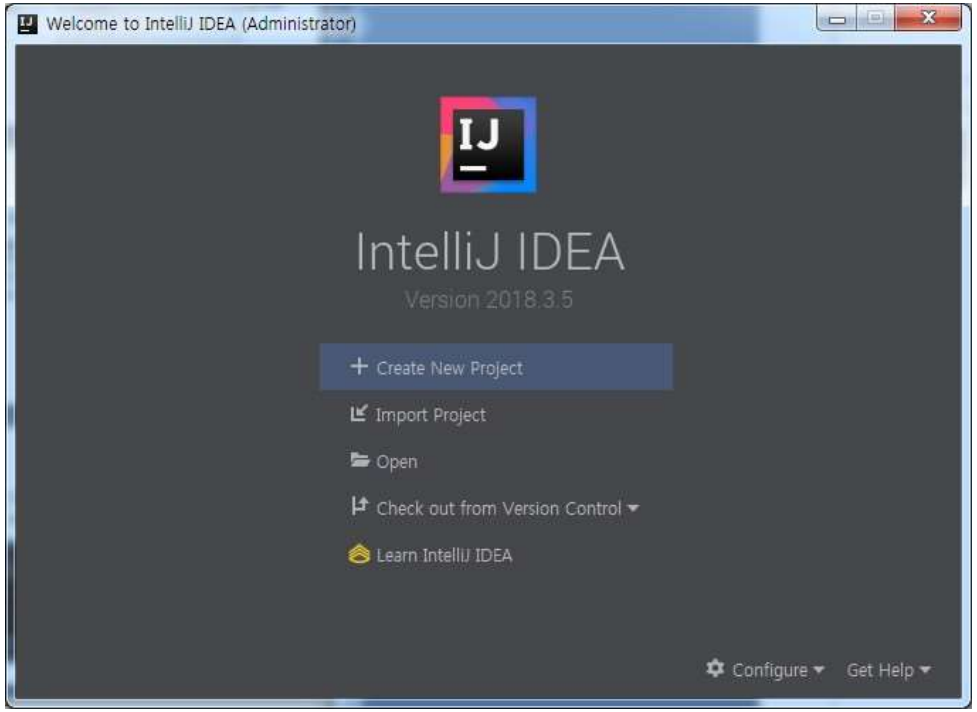
Installation 4

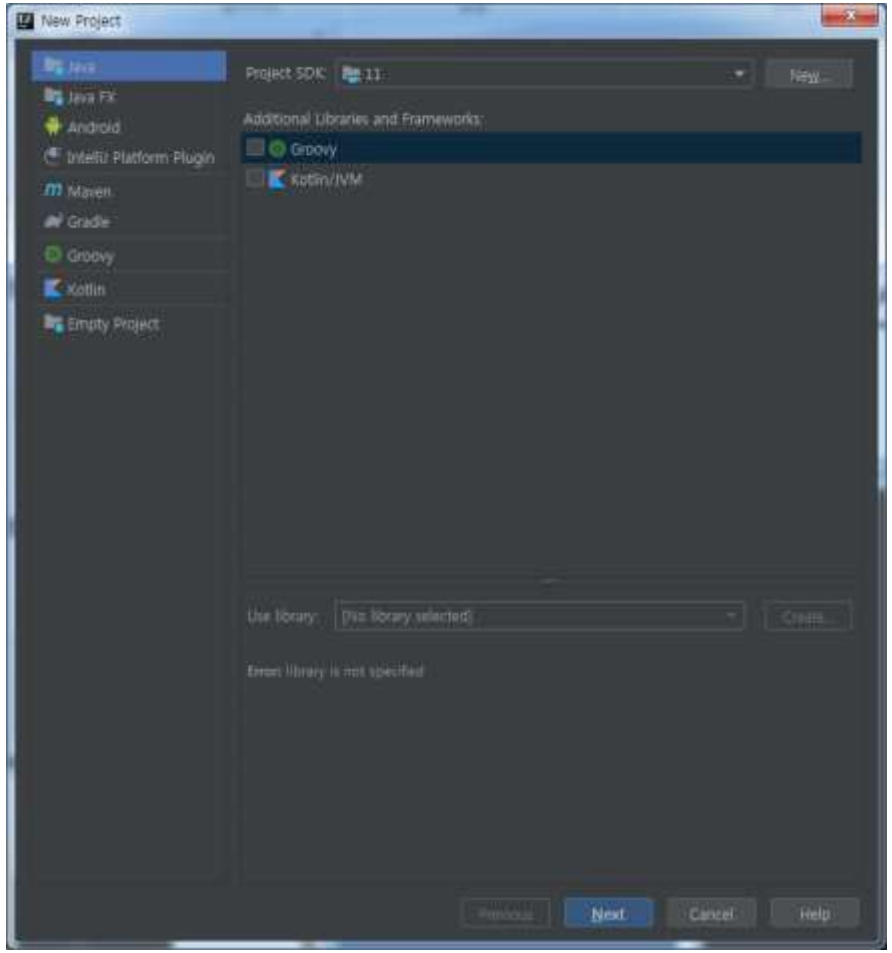
- Select either Darcula / Light
- Press Next -> Finished.



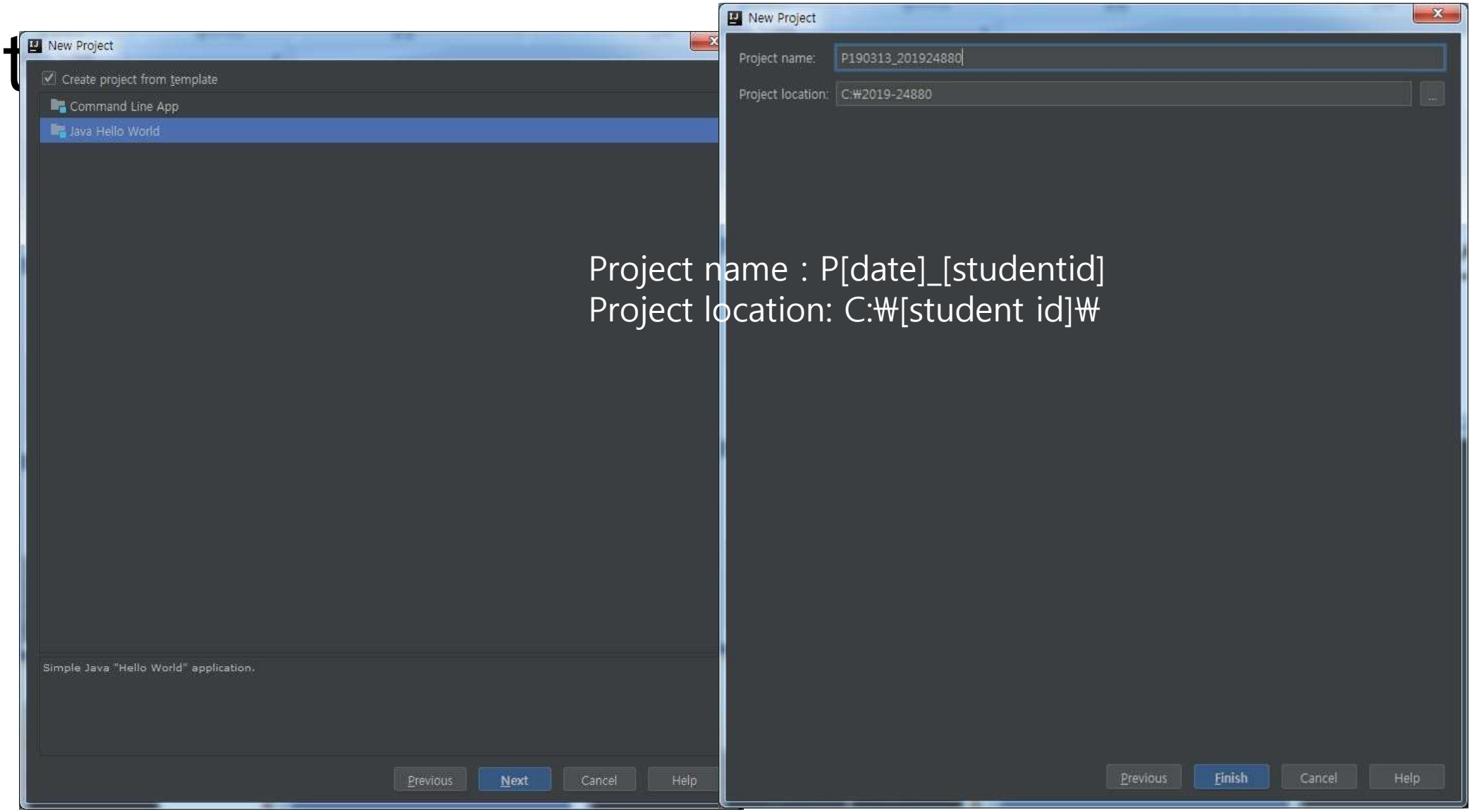


First Start-up settings



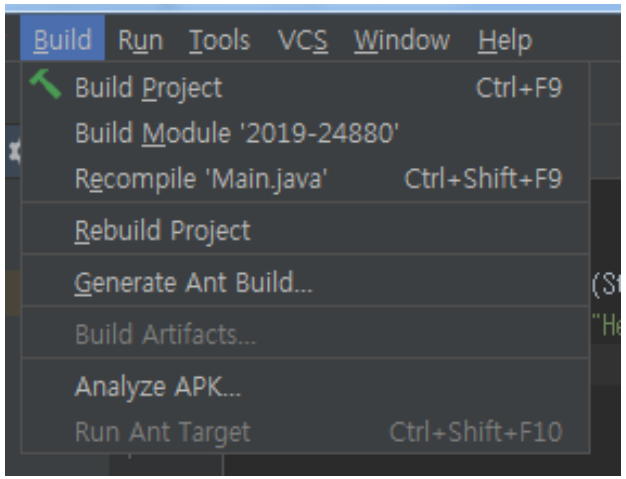


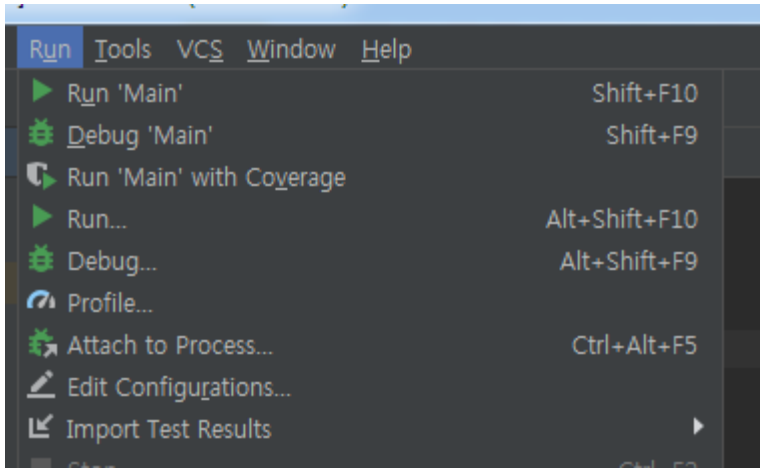
Cont



Project name : P[date]_[studentid]
Project location: C:₩[student id]₩

First Build and first run





Build (CTRL+F9)-> Run (SHIFT + F10)

Result :

```
"C:₩Program Files₩Java₩jdk-11.0.1₩bin₩java.exe"
Hello World!

Process finished with exit code 0
```

Example 1

Example1

```
public class HelloJava {
  public static void main(String args[]) {
     System.out.println("Hello, World");
  }
}
```

Example 2

Example2

```
class Item {
        public String title;
        public double price;
        private int inStockQuantity;
        public double SalePrice(){ return (price * 0.9);}
        public boolean isAvailable(){
                if(inStockQuantity > 0) return true;
                else return false;
        }

  public static void main(String args[]) {
                Item A = new Item();
            A.title = "comp";
            A.price = 1000;
            System.out.println(A.SalePrice());
  }
}
```

Example 3

Main.java

```java
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter username : ");

        String userName = scanner.nextLine();
        System.out.println("Username is: " + userName);
    }
}
```

Example 3

Input Types

In the example above, we used the `nextLine()` method, which is used to read Strings. To read other types, look at the table below:

Method	Description
`nextBoolean()`	Reads a `boolean` value from the user
`nextByte()`	Reads a `byte` value from the user
`nextDouble()`	Reads a `double` value from the user
`nextFloat()`	Reads a `float` value from the user
`nextInt()`	Reads a `int` value from the user
`nextLine()`	Reads a `String` value from the user
`nextLong()`	Reads a `long` value from the user
`nextShort()`	Reads a `short` value from the user

Example 4

Main.java

```java
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter username : ");
        String userName = scanner.nextLine();

        System.out.println("Enter age :");
        int age = scanner.nextInt();


        System.out.println("Username is: " + userName + ", and age is : "+age);
    }
}
```

Example 5

BufferedReader : 보통 한 줄씩 읽어올 때 사용합니다.
Scanner 보다 처리속도가 빨라서 보통 속도가 중요할 때 사용합니다.
(Ex. 백준 알고리즘 코딩 테스트)

Main.java

```java
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;

public class Main {

    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        int a = Integer.parseInt(br.readLine());
        System.out.println(a);
        br.close();
    }
}
```