

C++ Comments

 programiz.com/cpp-programming/comments

In this tutorial, we will learn about C++ comments, why we use them, and how to use them with the help of examples.

C++ comments are hints that a programmer can add to make their code easier to read and understand. They are completely ignored by C++ compilers.

There are two ways to add comments to code:

`//` - Single Line Comments

`/* */` -Multi-line Comments

Single Line Comments

In C++, any line that starts with `//` is a comment. For example,

```
// declaring a variable
int a;

// initializing the variable 'a' with the value 2
a = 2;
```

Here, we have used two single-line comments:

- `// declaring a variable`
- `// initializing the variable 'a' with the value 2`

We can also use single line comment like this:

```
int a;    // declaring a variable
```

Multi-line comments

In C++, any line between `/*` and `*/` is also a comment. For example,

```
/* declaring a variable
to store salary to employees
*/
int salary = 2000;
```

This syntax can be used to write both single-line and multi-line comments.

Using Comments for Debugging

Comments can also be used to disable code to prevent it from being executed. For example,

```
#include <iostream>
using namespace std;
int main() {
    cout << "some code";
    cout << 'error code;
    cout << "some other code";

    return 0;
}
```

If we get an error while running the program, instead of removing the error-prone code, we can use comments to disable it from being executed; this can be a valuable debugging tool.

```
#include <iostream>
using namespace std;
int main() {
    cout << "some code";
    // cout << 'error code;
    cout << "some other code";

    return 0;
}
```

Pro Tip: Remember the shortcut for using comments; it can be really helpful. For most code editors, it's `Ctrl + /` for Windows and `Cmd + /` for Mac.

Why use Comments?

If we write comments on our code, it will be easier for us to understand the code in the future. Also, it will be easier for your fellow developers to understand the code.

Note: Comments shouldn't be the substitute for a way to explain poorly written code in English. We should always write well-structured and self-explanatory code. And, then use comments.

As a general rule of thumb, use comments to explain **Why** you did something rather than **How** you did something, and you are good.