

Writing your first program

Follow these steps, copying the code exactly as it is listed, to create a simple C++ program that will output the traditional first program greeting:



hello.cpp



Comments throughout this book are shown in green – to differentiate them from other code.



After typing the final closing `}` brace of the main method, always hit Return to add a newline character – your compiler may insist that a source file should end with a newline character.

- 1 Open a plain text editor, such as Windows' Notepad, then type these “preprocessor directives”
`#include <iostream>`
`using namespace std ;`
- 2 A few lines below the preprocessor directives, add a “comment” describing the program
`// A C++ Program to output a greeting.`
- 3 Below the comment, add a “main function” declaration to contain the program statements
`int main()`
`{`

`}`
- 4 Between the curly brackets (braces) of the main function, insert this output “statement”
`cout << "Hello World!" << endl ;`
- 5 Next, insert a final “return” statement in the main function
`return 0 ;`
- 6 Save the program to any convenient location as “hello.cpp” – the complete program should look like this:

```
hello.cpp - Notepad
File Edit Format View Help
#include <iostream>
using namespace std ;

// A C++ Program to output a greeting.

int main()
{
    cout << "Hello World!" << endl ;
    return 0 ;
}
|
```

...cont'd

The separate parts of the program code on the opposite page can be examined individually to understand each part more clearly:

- **Preprocessor Directives** – these are processed by the compiler before the program code, so must always appear at the start of the page. Here, the **#include** instructs the compiler to use the standard C++ input/output library named **iostream**, specifying the library name between **< >** angled brackets. The next line is the “using directive” that allows functions in the specified namespace to be used without their namespace prefix. Functions of the **iostream** library are within the **std** namespace – so this **using** directive allows functions such as **std::cout** and **std::endl** to be simply written as **cout** and **endl**.
- **Comments** – these should be used to make the code more easily understood by others, and by yourself when revisiting the code later. In C++ programming, everything on a single line after a **//** double-slash is ignored by the compiler.
- **Main function** – this is the mandatory entry point of every C++ program. Programs may contain many functions, but they must always contain one named **main**, otherwise the compiler will not compile the program. Optionally, the parentheses after the function name may specify a comma-separated list of “argument” values to be used by that function. Following execution, the function must return a value to the operating system of the data type specified in its declaration – in this case, an **int** (integer) value.
- **Statements** – these are the actions that the program will execute when it runs. Each statement must be terminated by a semi-colon, in the same way that English language sentences must be terminated by a period (full stop). Here, the first statement calls upon the **cout** library function to output text and an **endl** carriage return. These are directed to standard output by the **<<** output stream operator. Notice that text strings in C++ must always be enclosed within double quotes. The final statement employs the C++ **return** keyword to return a zero integer value to the operating system – as required by the main function declaration. Traditionally, returning a zero value indicates that the program executed successfully.



The C++ compiler also supports multiple-line C-style comments between **/*** and ***/** – but these should only ever be used in C++ programming to “comment-out” sections of code when debugging.



Notice how the program code is formatted using spacing and indentation (collectively known as whitespace) to improve readability. All whitespace is ignored by the C++ compiler.