**Structure of a Program**

BASIC OF C++

Probably the best way to start learning a programming language is by writing a program. Therefore, here is our first program:

|  |  |
| --- | --- |
| **Input** | **Output** |
| // my first program in C++  #include <iostream>  using namespace std;  int main ()  {  cout << "Hello World!" ;  return 0;  } | Hello World! |

We are going to look line by line at the code we have just written:

**// my first program in C++**

* (**//**) As comment

**#include <iostream>**

* Lines beginning with a hash sign (#) are directives for the preprocessor.
* In this case the directive #include <iostream> tells the preprocessor to include the iostream standard file.

**using namespace std;**

* All the elements of the standard C++ library are declared within what is called a namespace, the namespace with the name std.

**int main ()**

* This line corresponds to the beginning of the definition of the main function. The main function is the point by where all C++ programs start their execution, independently of its location within the source code.
* The word ***main*** is followed in the code by a pair of parentheses (**()**).
* Right after these parentheses we can find the body of the main function enclosed in braces (**{}**).

**cout << "Hello World!" ;**

* This line is a C++ statement. A statement is a simple or compound expression that can actually produce some effect.
* ***cout*** represents the standard output stream in C++, and the meaning of the entire statement is to insert a sequence of characters (in this case the **Hello World** sequence of characters) into the standard output stream (which usually is the screen).
* ***cout*** is declared in the ***iostream*** standard file within the ***std*** namespace.
* Notice that the statement ends with a semicolon character (**;**). This character is used to mark the end of the statement and in fact it must be included at the end of all expression statements in all C++ programs.

**return 0;**

* The ***return*** statement causes the main function to finish.
* A ***return*** code of 0 for the main function is generally interpreted as the program worked as expected without any errors during its execution.

The program has been structured in different lines in order to be more readable, but in C++, we do not have strict rules on how to separate instructions in different lines. For example, instead of :

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

We could have written:

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

**All in just one line and this would have had exactly the same meaning as the previous code.**

Let us add an additional instruction to our first program:

|  |  |
| --- | --- |
| // my second program in C++  #include <iostream>  using namespace std;  int main ()  {  cout << "Hello World!" ;  cout << "I'm a C++ program" ;    return 0;  } | Hello World! I’m a C++ program |

We could have written:

|  |
| --- |
| int main () { cout << “Hello World!” ; cout << “I’m a C++ program” ; return 0; } |

We were also free to divide the code into more lines if we considered it more convenient:

|  |
| --- |
| int main ()  {  cout <<  “Hello World!” ;  cout  << “I’m a C++ program” ;    return 0;  } |

**And the result would again have been exactly the same as in the previous examples.**