

Hello world explained  
author: andrew ycas

so last time we got the compiler workin, but what good is a compiler if u cant code..

```
--  
#import "iostream.h"  
#import "stdio.h"  
  
int main()  
{  
  
    std::cout << "hello world" ;  
    getchar();  
    return 0;  
}  
--
```

above is the code i gave you last time, and it all may be confusing, it may not.

But in this lesson it will all be explained.

If u read the terms u should know part, u know that the commands that are programmed are in a library. Well you have to tell the computer which libraries you are talking about, the #import or #include statement does this.( i didn't use #include cause i forgot about it, thats how you are supposed to do it, but i forget when i wrote the previous lesson.. its all java's fault) In this program we are telling the computer that we wanna use the stdio and the iostream librarys.

Next in the code is int main(). You need main in every c++ program you write, and in most other languages u need a main. Main the function, or the list of code, that will be run when u run the program itself. We will talk about having other functions besides main later, but for right now u just need to know that u have to have a main, and that main is function. The "int" before the main just means that this function will return a variable. Functions will all be explained later, but that bit of information u need to know for main right now.

The two brackets are just signifying what code is in the main function. If u had no brackets, the computer would only associate the line right under the main declaration with main, and the computer would think is just floating out there (and in this case if u had no brackets u would get compile errors).

Next is the std::cout << "hello world" ; command. Now u can probably guess what this is doing, it is telling the computer to display hello world on the screen. Notice that this line ends in a semi colon. This tells the computer that it has reached the end of the line of this command. Now not all lines need a semicolon at the end, for example, there is no semicolon at the end of main, but right now just know that actual commands need semicolons, and other stuff doesnt.

When i talk about commands i mean when u tell the computer to do something, within a function. So u would not use a semicolon when you would include a library, but when u tell the computer to display something using std::cout, then u would. If you are confused, don't worry, i cant explain this too well, but it is just one of those things that you will pick up while you are codeing. And usually if there is a compile error, u forgot a semicolon at the end of a line.

The next command, `getchar()` ; is like `std::cout` in that it is a command and need a semicolon at the end. What `getchar()` does is that it waits for the user to press a key on the keyboard before continuing on through the program. Try compiling and running without `getchar()` and ull see that when the program runs that it flashes up on the screen and then goes away. Now, if ur program is called from the command prompt, then it wont flash up on the screen. But we wont get into the command prompt right now, but if u know how to use it go ahead and launch ur program without `getchar()`, and u will see that the window doesnt close, and helloworld is still displayed.

The last command is the return statement. This code has to deal with the fact that main is a function, and it will be hit on in more detail later, just know that right now u need it as the last line of main.

And that is the end of helloworld. I hope that this was explained well enough, if i didn't explain something well enough then just ask me and ill answer ur questions...

next lesson: variables and cin ..