

Hello world++

Programação (L.EIC009)

Eduardo R. B. Marques, DCC/FCUP

Hello world! ++

Exemplo - hello.cpp:

```
/*  
    A simple program that prints "Hello world! ++"  
*/  
  
#include <iostream>  
  
int main() {  
    // Print "Hello world! ++" ...  
    std::cout << "Hello world! ++" << std::endl;  
    return 0;  
}
```

Compilação com GCC:

```
g++ -Wall hello.cpp -o hello
```

Execução:

```
$ ./hello
```

```
Hello world! ++
```

Aspectos gerais

- A extensão `.cpp` é normalmente usada para ficheiros com código C++. Também é relativamente comum empregar `.cxx` ou `.cc`.
- Header files por sua vez são definidas com extensão `.h` ou `hpp`, mas também sem qualquer extensão como percebemos no exemplo dado que contém `#include <iostream>`.
- Código C pode de forma geral ser usado sem modificações e com a mesma semântica em C++ (ou falta dela no que toca a comportamento indefinido).

“Mistérios”

Linha de código “misteriosa”:

```
std::cout << "Hello world! ++" << std::endl;
```

Várias perguntas:

- O que é `std::cout` ?
- O que é `std`?
- Como é que está a ser usado o operador `<<` ?
- O que é `std::endl` ?

Respostas (parciais) a seguir ...

`std::cout ?`

```
std::cout << "Hello world! ++" << std::endl;
```

O que é `std::cout` ?

Em C++ podemos usar **objectos** que são instâncias de **classes**.
Temos um primeiro exemplo disso.

`std::cout` é um **objecto** declarado globalmente via header `iostream` que serve para escrever para o output standard do programa, analogamente a `stdout` em C (que usamos implicitamente por ex. numa chamada a `printf`).

`std::cout` é uma instância da **classe** `std::ostream` que define várias funções que podemos usar em associação a um objecto desse tipo.

std ?

```
std::cout << "Hello world! ++" << std::endl;
```

O que é std?

É o “namespace” (espaço de nomes) para a declaração de `cout` e várias outras declarações “standard” da biblioteca standard de C++.

Declarações (de tipos, classes, funções, ...) em C++ podem ser associadas a “name spaces”.

A função `main` no exemplo está associada ao **namespace raiz** que não tem um nome associado.

Uso de operador << ?

```
std::cout << "Hello world! ++" << std::endl;
```

Como é que está a ser usado o operador << ?

O uso de << corresponde à invocação de uma **função que define o significado do operador <<** para o tipo `int`.

```
ostream& operator<<( int value );
```

Na verdade podemos encontrar várias definições para o operador << com diferentes tipos associados, ex.

```
ostream& operator<< (bool val);  
ostream& operator<< (float val);  
ostream& operator<< (double val);
```

Esta facilidade de C++ é conhecida por **“operator overloading”**. Veremos depois como funciona em mais detalhe.

`std::endl` ?

```
std::cout << "Hello world! ++" << std::endl;
```

O que é `std::endl` ?

É um manipulador de input/output que serve para causar a escrita de uma mudança de linha (`'\n'`) e “flushing” do output de saída.