Introduction

C++ is a strongly typed, compiled, high performance language, which evolved from c. It shares c's syntax, as well as c's ability to interface fairly directly with hardware. It can be used to write very "close to the metal" code, as well as highly abstracted code. C++ places more responsibilities on the developer, and it is likely that the two areas which will be the most alien to you, the python develoepr, are memory management and strict typing. However, the big payoff in taking these responsibilities on is the ability (notice i didnt say guarantee) of producing high performance code.

structure of a c++ program

Like a python program, a C++ program can consist of many files. And like a python program, a c++ executable ultimately has a single entry point. In C++'s case that entry point is the **main function**. So let's write our first c++ program. Create a file called *hello.cpp* with the following in it:

```
#include <iostream>

int main() {
    std::cout << "hello world" << std::endl;
    return 0;
}
```

And compile it on the command line:

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

Tada, you should have an executable called hello, which will greet the world when run. Even in this trivial program, we have a lot to talk about. So let's get at it.

includes, the modules of the python world

In c++, like in python, you can split your implementation of a program up into multiple files. And like, python, you can leverage existing libraries to do much of the low level work for you; you are not stuck writing everything from scratch, and much of what is considered modern C++ is actually provided by the Standard Library, and not the language itself (although the standard library is, well standard, and available on all compilers so...)

In python, you import modules. C++ doesnt have a module system (yet. its on the way).However, you can include other code using the #include preprocessor directive. There are actually two forms:

```
#include<>
```

and

```
#include ""
```

The <> form is used to include libraries which are provided externally. For instance, C++ ships with a large library, much like the standard library python ships with.

The “” form is used to include other files which are in your project. These are similar to the internal import statements in a python package.

The directive

```
#include <iostream>
```

imports the iostream library which makes a number of functions and classes available to us in the `std` namespace. (we will cover namespaces later). A namespace is a label used to disambiguate names and avoid clashes. Namespaces prefix lables and are affixed using ‘::’. For example `std::cout` from above references *cout* in the *std* namespace.