Standard and Non–Standard Libraries in C/C++

February 2, 2012

1 Standard Libraries in C

By convention, libraries in C have a .h extension. For a complete listing of ANDI–ISO standard libraries in C please see [1]. As an example, *conio.h* is not part of the standard C and isn't guaranteed to be available or work with all compilers.

2 Standard Libraries in C++

Standard libraries in C++ do not have any extension, e.g. *iostream*, *fstream*. Note that *iostream**.h*** is not a standard C++ header. Standard C header files in C++ normally start with a 'c' and without .h extension. For example *cstdio*, *cstdlib* etc. For a complete listing of standard C++ libraries see [2].

3 `namespace std` in C++

Following OOP concepts, functions and variables in C++ headers are not part of the global space and are enclosed in *namespace std*. Following comparison illustrates this:

iostream	iostream.h
`namespace std`	`cout`
`{`	`cin`
`cout`	`endl`
`cin`	`...`
`endl`	`...`
`...`	`...`
`}`	

Members of `std` namespace can be accessed with the scope resolution operator, e.g. `std::cin`, `std::cout` etc. A `using` directive can be used to access a particular function or object without the need to use scope resolution operator with every usage, e.g. `using std::cin;`. With a `using namespace std;` directive, all objects and functions within the namespace become globally accessible and this is considered a bad programming practice.

4 The `string` class in C++

The standard header for *string.h* in C++ is *cstring*. Another header *string* is available in C++ and contains the implementation of `string` class. This is completely different from the *string.h* and *cstring* libraries.

References

[1] `http://en.wikipedia.org/wiki/C_standard_library`.

[2] `http://en.wikipedia.org/wiki/C_%2B_%2B_standard_library`.