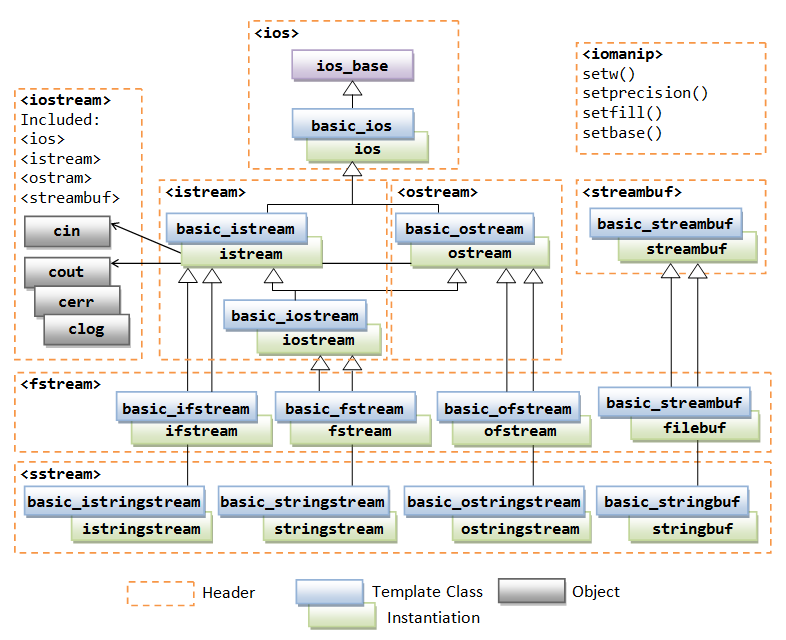
**What is #include<iostream>?**

* **iostream** is a class in the **std** namespace that abstracts all the behaviors of input and output streams. It is not a “flat” class. It is really the end result of a multiple-inheritance chain that includes the classes **ios** the base class of **istream** and **ostream**, which in turn are the base classes of **iostream**. In addition, it provides implementations for such useful objects as **cout**, **cin**, and **cerr**.
* <iostream> is a header file. This file defines the cin, cout, cerr and clog objects, which corresponds to the standard input stream, the standard output stream, the un-buffered standard error stream, and the buffered standard error stream, respectively.

To fully understand **iostream**, in addition to the object-oriented paradigm the programmer has to be familiar with the concept of streams, multiple inheritance and directed acyclic graphs (DAGs)—Understanding DAGs is needed to see why and where virtual inheritance is required to avoid violating the ***one-definition rule—***. In short, to be proficient a C++ programmer must learn and study the C++ language very well. The resulting superior knowledge of such programmer significantly increases the quality of the code(s) he produces.

The C/C++ languages require skills not needed in other computer languages. Therefore, I advise prospective users of the language that one **should not** claim expertise in C/C++ programming simply because one refactored into C/C++ programs written in other languages.



What does namespace means in c++ program?

A **namespace** is a declarative region that provides a scope to the identifiers (the names of types, functions, variables, etc) inside it. Namespaces are used to organize code into logical groups and to prevent name collisions that can occur especially when your code base includes multiple libraries. All identifiers at namespace scope are visible to one another without qualification. Identifiers outside the namespace can access the members by using the fully qualified name for each identifier, for example std::vector<std::string> vec;, or else by a [using Declaration](https://docs.microsoft.com/en-us/cpp/cpp/using-declaration?view=vs-2017) for a single identifier (using std::string), or a [using Directive](https://docs.microsoft.com/en-us/cpp/cpp/namespaces-cpp?view=vs-2017#using_directives) for all the identifiers in the namespace (using namespace std;). Code in header files should always use the

What does int (main) means?

The purpose of **main** 's return value is to return an exit status to the operating system. In standard C, the only valid signatures for **main** are: **int main**(void) and **int main**(**int** argc, char \*\*argv) The form you're using: **int main()** is an old style declaration that indicates **main** takes an unspecified number of arguments.