C++ ENVIRONMENT SETUP

http://www.tutorialspoint.com/cplusplus/cpp_environment_setup.htm

> ### *Try it Option Online*
>
> *You really do not need to set up your own environment to start learning C++ programming language. Reason is very simple, we already have set up C++ Programming environment online, so that you can compile and execute all the available examples online at the same time when you are doing your theory work. This gives you confidence in what you are reading and to check the result with different options. Feel free to modify any example and execute it online.*
>
> *Try the following example using **Try it** option available at the top right corner of the below sample code box:*
>
> ```
> #include <iostream>
> using namespace std;
>
> int main()
> {
>    cout << "Hello World";
>    return 0;
> }
> ```
>
> *For most of the examples given in this tutorial, you will find **Try it** option, so just make use of it and enjoy your learning.*

Local Environment Setup

If you are still willing to set up your environment for C++, you need following two softwares available on your computer.

Text Editor:

This will be used to type your program. Examples of few editors include Windows Notepad, OS Edit command, Brief, Epsilon, EMACS, and vim or vi.

Name and version of text editor can vary on different operating systems. For example, Notepad will be used on Windows and vim or vi can be used on windows as well as Linux, or UNIX.

The files you create with your editor are called source files and for C++ they typically are named with the extension .cpp, .cp, or .c.

Before starting your programming, make sure you have one text editor in place and you have enough experience to type your C++ program.

C++ Compiler:

This is actual C++ compiler, which will be used to compile your source code into final executable program.

Most C++ compilers don't care what extension you give your source code, but if you don't specify otherwise, many will use .cpp by default

Most frequently used and free available compiler is GNU C/C++ compiler, otherwise you can have compilers either from HP or Solaris if you have respective Operating Systems.

Installing GNU C/C++ Compiler:

UNIX/Linux Installation:

If you are using **Linux or UNIX** then check whether GCC is installed on your system by entering the following command from the command line:

```
$ g++ -v
```

If you have installed GCC, then it should print a message such as the following:

```
Using built-in specs.
Target: i386-redhat-linux
Configured with: ../configure --prefix=/usr .......
Thread model: posix
gcc version 4.1.2 20080704 (Red Hat 4.1.2-46)
```

If GCC is not installed, then you will have to install it yourself using the detailed instructions available at http://gcc.gnu.org/install/

Mac OS X Installation:

If you use Mac OS X, the easiest way to obtain GCC is to download the Xcode development environment from Apple's web site and follow the simple installation instructions.

Xcode is currently available at developer.apple.com/technologies/tools/.

Windows Installation:

To install GCC at Windows you need to install MinGW. To install MinGW, go to the MinGW homepage, www.mingw.org, and follow the link to the MinGW download page. Download the latest version of the MinGW installation program which should be named MinGW-<version>.exe.

While installing MinWG, at a minimum, you must install gcc-core, gcc-g++, binutils, and the MinGW runtime, but you may wish to install more.

Add the bin subdirectory of your MinGW installation to your **PATH** environment variable so that you can specify these tools on the command line by their simple names.

When the installation is complete, you will be able to run gcc, g++, ar, ranlib, dlltool, and several other GNU tools from the Windows command line.