

## Switching from Turbo C++ to g++

Turbo C++ users, please note that Turbo C++ uses a v. old 16-bit compiler which is outdated and is not in compliance with the latest standards, while the latest compilers are 32-bit or 64-bit and comply to the standards. If you haven't already, it's the right time to switch from TurboC++ to the worldwide used standard compiler like **gcc/g++** or **mingw-g++/mingw-gcc**.

While gcc is extensively used in Linux, mingw-gcc is the Windows version of gcc. Although it's best to code in Linux, but if you're more comfortable with Windows, then I'll recommend you to use **CodeBlocks** IDE (available at [www.codeblocks.org](http://www.codeblocks.org)) which uses Mingw-gcc as the compiler. CodeBlocks is available for both Windows and Linux, and supports features like code completion, braces completion, etc. being very easy-to-use. You can of course use any of the other available IDE(s), but just make sure that the one you choose uses/supports gcc or mingw-gcc.

Alternatively, you can code without using any IDE too in Linux. You can use any editing tools like gedit, vim, nano, etc. to write code just like you write other text files using them (and of course give the extension as .cpp (or .c) to them). And to compile and run, you can enter the following (simplest) commands in the terminal:

To compile: `g++ file-name.cpp`

To run: `./a.out`

If your code includes math header file, then compile with `-lm` option.

No programming contest (except some school contests) and no online coding platforms (like CodeChef, Spoj, etc.) supports Turbo C++, but use gcc for compilation. To make the code compilable with gcc and to ensure that the latest standards are met, few minor changes are required in the Turbo C++ code (though the rest of the code remains the same).

Here are some tips:

- There's no header file named as '**conio.h**' in gcc. Including such header file would result in compilation error. Since conio.h header file doesn't exist, the latest standard does not support functions like `clrscr()`, `getch()`, `getche()`, `gets()`, `cgets()`, `puts`, `putch()`, `cputs()`, etc. too. Hence, using such functions would also result in compilation error. So, do NOT use conio.h header file and any of its functions. So how do you clear the screen then? You don't need to. At least you won't be required to clear the screen in the programming contests (whenever the program runs/starts, the output stream (if not the screen) is cleared automatically, and the contest's judging system compares the output in the output stream only with the expected output). Isn't `getch()` a necessary function to use? No! Program can v.well work (and generate output) without that function. If `gets()` is not allowed, how do you then take string as input? You can always input strings using `cin` or `scanf()`. You would generally not be required to take strings with spaces as input in any programming contest. And if you ever have to, you can use functions like `cin.getline()`, etc. In short, you can always do without using `conio.h` header file and any of its functions.
- A little more about including header files in g++: Functions from (most of) rest of the header files can be used normally, as you did in Turbo. But some of the header files are included differently. For example, you write `#include<iostream>` instead of `#include<iostream.h>` //Not accepted Same goes for `iomanip.h`. You include `iomanip` instead. And header files which were originally a part of C language (like `math.h`, `string.h`, `stdio.h`, `time.h`, `stdlib.h`, `ctype.h`, etc.) must have 'c' as a prefix to them (and shouldn't be having .h at the end). You include them like:  
`#include<cmath>` `#include<cstring>` `#include<cstdlib>` `#include<ctime>`  
`#include<cstdlib>` `#include<cctype>`
- Either you prefix `std::` with all `cout` and `cin` statements and few other places, or we'd advice you to add the following line outside the main function: `using namespace std;`
- Do not use `void main()`, and only `int main()`, and always `return 0` at the end of the main() function.

Here's how a 'Hello World' program in gcc/g++ look like:

```
#include<iostream> using namespace std; int main() { cout<<"Hello IIIT-ians!\n"; return 0; }
```

Hoping that this short tutorial would help you in shifting from Turbo C++ to gcc/g++. I'd advice you to solve few easy problems at CodeChef or other such platforms using gcc compiler to be confident about the shift. Happy Coding! :)

- Harsh Manocha  
BTech 2012