Compiling a simple C program using GCC

How to compile a simple C program using the **gcc** compiler:

Let's try with the simple Hello World program in C language which is given as:

```
#include <stdio.h>
int main (int argc, char **argv)
{
     printf ("Hello, world!\n");
     return 0;
}
```

We will save the above C code in a file named **hello.c**. To compile a ".c" extension file with gcc, use the following command:

```
% gcc -Wall hello.c -o hello
```

This compiles the source code in **hello.c** to machine code and stores it in an executable file named **hello**.

NOTE: The output file for the machine code is specified using the **-o** option. This option is usually given as the last argument on the command line. If it is omitted, the output is written to a default file called **a.out**. If a file with the same name as the executable file already exists in the current directory, it will be overwritten.

The option **-Wall** turns on all the most commonly-used compiler warnings so It is recommended that you always use this option! There are many other warning options, but -Wall is the most important. GCC will not produce any warnings unless they are enabled. Compiler warnings are an essential aid in detecting problems when programming in C and C++. In this case, the compiler does not produce any warnings with the -Wall option, since the program is completely valid. Source code which does not produce any warnings is said to compile cleanly.

To run the program, type the path name of the executable as below;

```
% ./hello
```

This loads the executable file into memory and causes the CPU to begin executing the instructions contained within it. The path **./** refers to the current directory, so **./hello** loads and runs the executable file 'hello' located in the current directory.