



LEARN C PROGRAMMING

c programming language

tutorialspoint

SIMPLY EASY LEARNING

1. Environment setup

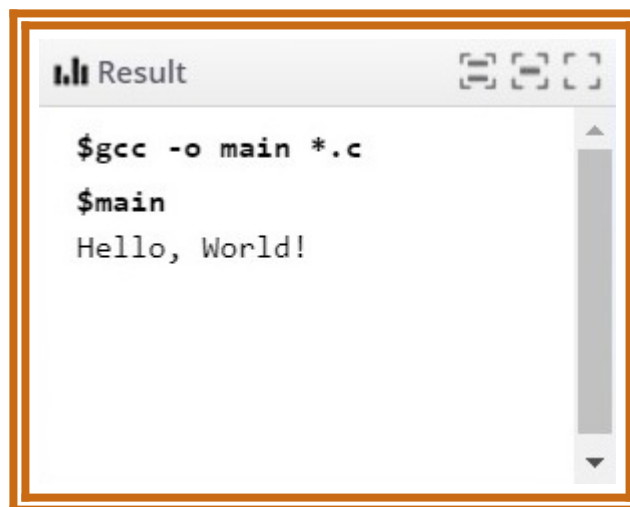
Try it Online

! 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 following example using our online compiler option available at <http://www.compileonline.com/>.

```
#include <stdio.h>

int main()
{
    /* my first program in C */
    printf("Hello, World! \n");
    return 0;
}
```

* For most of the examples given in this tutorial, you will find the **Try it** option in our website code sections at the top right corner that will take you to the online compiler. So just make use of it and enjoy your learning.

A screenshot of a terminal window titled "Result" with a scrollbar on the right. The terminal shows the following text:

```
$gcc -o main *.c
$main
Hello, World!
```

Local Environment Setup

? If you want to set up your environment for C programming language, you need the following two software tools available on your computer: Text Editor and The C Compiler.

Installation on UNIX/Linux

? 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:

```
$ gcc -v
```

? If you have GNU compiler installed on your machine, then it should print a message as follows:

```
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 gcc.gnu.org/install/.

Installation on Mac OS

? 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. Once you have Xcode setup, you will be able to use GNU compiler for C/C++.

Installation on Windows

? To install GCC on 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 MinGW, 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. After 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.

2. Basic syntax

Identifiers

! A C identifier is a name used to identify a variable, function, or any other user- defined item. An identifier starts with a letter A to Z, a to z, or an underscore ‘_’ followed by zero or more letters, underscores, and digits (0 to 9).

! C does not allow punctuation characters such as @, \$, and % within identifiers. C is a **case-sensitive** programming language. Thus, *Customer* and *customer* are two different identifiers in C. Here are some examples of acceptable identifiers:

```
Zara
abc
move_name
a_123
myname50
_temp
j
retVal
```

Keywords

! The following list shows the reserved words in C. These reserved words **may not be used** as constants or variables or any other identifier names. While naming your functions and variables, other than these names, you can choose any names of reasonable length for variables, functions etc.

Id	Keyword	Description
1	auto	The auto keyword declares automatic variables
2	break	The break statement makes program jump out of the innermost enclosing loop (while, do, for or switch statements) explicitly.
3	case	The switch and case statement is used when a block of statements has to be executed among many blocks
4	char	The char keyword declares a character variable.
5	const	An identifier can be declared constant by using const keyword
6	continue	The continue statement skips the certain statements inside the loop.
7	default	Used in switch

Id	Keyword	Description
8	do	Used in do...while loop
9	double	Keywords double and float are used for declaring floating type variables.
10	else	Decision making.
11	enum	Enumeration types are declared in C programming using keyword enum.
12	extern	The extern keyword declares that a variable or a function has external linkage outside of the file it is declared.
13	float	Keywords double and float are used for declaring floating type variables.
14	for	Used in a loop.
15	goto	Go to label.
16	if	Decision making.
17	int	Integer type.
18	long	Long integer type.
19	register	Creates register variables which are much faster than normal variables.
20	return	Exit from function, return value.
21	short	Short integer.
22	signed	Integer modifier.
23	sizeof	Gets a size.
24	static	Creates static variable. The value of the static variables persists until the end of the program.
25	struct	Structure.
26	switch	The switch and case statement is used when a block of statements has to be executed among many blocks
27	typedef	The typedef keyword is used to explicitly associate a type with an identifier.

Id	Keyword	Description
28	union	A Union is used for grouping different types of variable under a single name.
29	unsigned	Unsigned integer.
30	void	The void keyword indicates that a function doesn't return any value.
31	volatile	A volatile object can be modified in an unspecified way by the hardware.
32	while	Used in a loop.

Bitwise Operators

🚩 Bitwise operators work on bits and perform bit-by-bit operation. They are `&`, `|`, `^` and `~`.

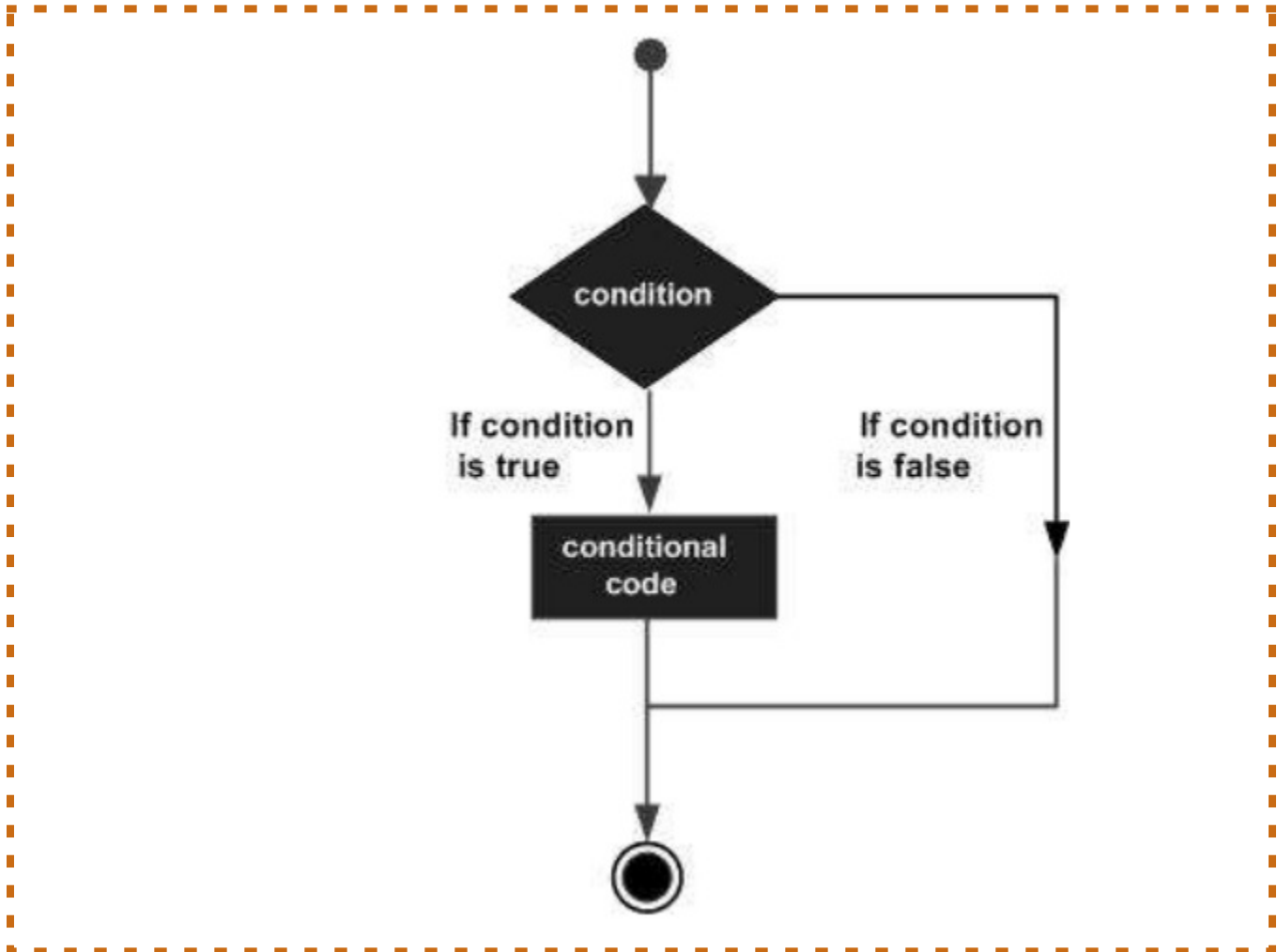
Assume A = 60 and B = 13; in binary format, they will be as follows:

```
A = 0011 1100
B = 0000 1101
A&B = 0000 1100
A|B = 0011 1101
A^B = 0011 0001
~A = 1100 0011
```

Typical structure

? Decision-making structures require that the programmer specifies one or more conditions to be evaluated or tested by the program, along with a statement or statements to be executed if the condition is determined to be true, and optionally, other statements to be executed if the condition is determined to be false.

Shown below is the general form of a typical decision-making structure found in most of the programming languages:



CONTENTS

LEARN C PROGRAMMING	1
1. Environment setup	2
Try it Online	2
Local Environment Setup	2
Installation on UNIX/Linux	2
Installation on Mac OS	3
Installation on Windows	3
2. Basic syntax	4
Identifiers	4
Keywords	4
Bitwise Operators	6
3. Decision making	7
Typical structure	7