**C Programming**

1. C is a powerful general-purpose programming language. It is fast, portable and available in all platforms.
2. C is a low level language and also it called as Structured Programming language.
3. If you are new to programming, C is a good choice to start your programming.
4. how to get started in C programming language, why you should learn it and how you can learn it.

* [**What is C Programming?**](https://www.programiz.com/c-programming#what-is-c)
* [**Your First C Program**](https://www.programiz.com/c-programming#first-c-program)

**What is C (Programming Language)?**

Before getting started with C programming, let’s get familiarized with the language first.

C is a general-purpose programming language used for wide range of applications from Operating systems like Windows and iOS to software that is used for creating 3D movies.

C programming is highly efficient. That’s the main reason why it’s very popular despite being more than 40 years old.

Standard C programs are portable. The source code written in one system works in another operating system without any change.

As mentioned, it’s a good language to start learning programming. If you know C programming, you will not just understand how your program works, but will also be able to create a mental picture on how a computer works.

### What will you gain if you learn C?

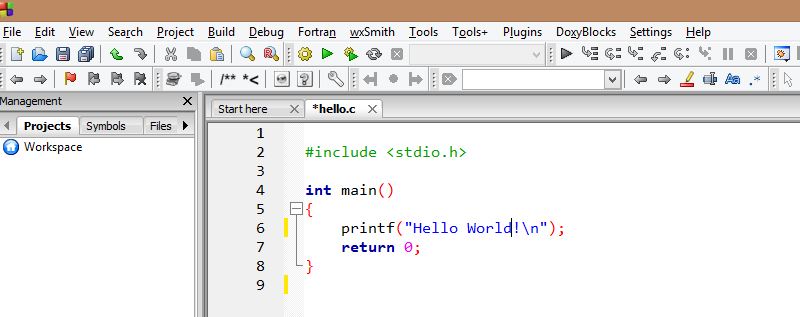
If you don’t know C, you don’t know what you are doing as a programmer. Sure, your application works fine and all. But, if you can’t say why while (\*s++ = \*p++); copies a string.ss

1. **You will understand how a computer works.**  
     
   If you know C, you will not only know how your program works but, you will be able to create a mental model on how a computer works (including memory management and allocation). You will learn to appreciate the freedom that C provides unlike Python and Java.  
     
   Understanding C allows you to write programs that you never thought were possible before (or at the very least, you will have a broader understanding of computer architecture and programming as a whole).

## Compile and run C programming on your OS

There are numerous compilers and text editors you can use to run C programming. These compilers and text editors may differ from system to system.

You will find the easiest way to run C programming on your computer (Windows, Mac OS X or Linux) in this section.



## The fun begins: Your first C program

You will learn to write a “Hello, World!” program in this section.

### Why “Hello, World!” program?

“Hello, World!” is a simple program that displays “Hello, World!” on the screen. Since, it’s a very simple program, it is used to illustrate the basic syntax of any programming language.

This program is often used to introduce programming language to a beginner. So, let’s get started.

#include <stdio.h>

int main()

{

printf("Hello, World!\n");

return 0;

}

### How “Hello, World!” program works?

**Include stdio.h header file in your program.**

C programming is small and cannot do much by itself. You need to use libraries that are necessary to run the program. The stdio.h is a header file and C compiler knows the location of that file. To use the file, you need to include it in your program using #include preprocessor.

**Why do you need stdio.h file in this program?**

In this program, we have used printf() function which displays the text inside the quotation mark. Since printf() is defined in stdio.h, you need to include stdio.h.

**The main() function**

In C programming, the code execution begins from the start of main() function (doesn’t matter if main()isn’t located at the beginning).

The code inside the curly braces { } is the body of main() function. The main() function is mandatory in every C program.

int main() {

}

This program doesn’t do anything but, it’s a valid C program.

**The printf() function**

The printf() is a library function that sends formatted output to the screen (displays the string inside the quotation mark). Notice the semicolon at the end of the statement.

In our program, it displays Hello, World! on the screen.

Remember, you need to include stdio.h file in your program for this to work.

**The return statement**

The return statement return 0; inside the main() function ends the program. This statement isn’t mandatory. However, it’s considered good programming practice to use it.

### Key notes to take away

* All C program starts from the main() function and it’s mandatory.
* You can use the required header file that’s necessary in the program. For example: To use sqrt()function to calculate square root and pow() function to find power of a number, you need to include math.h header file in your program.
* C is case-sensitive; the use of uppercase letter and lowercase letter have different meanings.
* The C program ends when the program encounters the return statement inside the main() function. However, return statement inside the main function is not mandatory.
* The statement in a C program ends with a semicolon.

### INTRODUCTION

### Tokens: The smallest individual elements or units in a program are called as Tokens. We cannot split it

### example C :main() will not write ad ma in ()

### C has following tokens:

1. **Keywords               (eg: int, while,if),**
2. **Identifiers               (eg: main, total),**
3. **Constants              (eg: 10, 20),**
4. **Strings                    (eg: “total”, “hello”),**
5. **Special symbols   (eg: (), {}),**
6. **Operators              (eg: +, /,-,\*)**

## 

## Character set

Character set is a set of alphabets, letters and some special characters that are valid in C language.

### Alphabets

Uppercase: A B C-.... X Y Z

Lowercase: a b c..... x y z

C accepts both lowercase and uppercase alphabets as variables and functions.

### Digits

0 1 2 3 4 5 6 7 8 9

### Special Characters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Special Characters in C Programming | | | | |
| , | < | > | . | \_ |
| ( | ) | ; | $ | : |
| % | [ | ] | # | ? |
| ' | & | { | } | " |
| ^ | ! | \* | / | | |
| - | \ | ~ | + |  |

## C Keywords

Keywords are predefined, reserved words used in programming that have special meanings to the compiler. Keywords are part of the syntax and they cannot be used as an identifier. For example:

int money;

Here, int is a keyword that indicates 'money' is a [variable](https://www.programiz.com/c-programming/c-variables-constants) of type integer.

As C is a case sensitive language, all keywords must be written in lowercase. Keywords in C Language:

|  |  |  |  |
| --- | --- | --- | --- |
| auto | double | int | struct |
| break | else | long | switch |
| case | enum | register | typedef |
| char | extern | return | union |
| continue | for | signed | void |
| do | if | static | while |
| default | goto | sizeof | volatile |
| const | float | short | unsigned |

Along with these keywords, C supports other numerous keywords depending upon the compiler.All these keywords, their syntax and application will be discussed in their respective topics. However, if you want a brief overview on these keywords without going further.

## C Identifiers

Identifier refers to name given to entities such as variables, functions, structures etc.

Identifier must be unique. They are created to give unique name to a entity to identify it during the execution of the program. For example:

int money;

double accountBalance;

Here, money and accountBalance are identifiers.

Also remember, identifier names must be different from keywords. You cannot use int as an identifier because int is a keyword.

### Rules for writing an identifier

1. A valid identifier can have letters (both uppercase and lowercase letters), digits and underscores.
2. The first letter of an identifier should be either a letter or an underscore. However, it is discouraged to start an identifier name with an underscore.
3. There is no rule on length of an identifier. However, the first 31 characters of identifiers are discriminated by the compiler.

## Constants/Literals

A constant is a value or an identifier whose value cannot be altered in a program. For example: 1, 2.5, "C programming is easy", etc.

As mentioned, an identifier also can be defined as a constant.

const double PI = 3.14

### 4. Escape Sequences

Sometimes, it is necessary to use characters which cannot be typed or has special meaning in C programming. For example: newline(enter), tab, question mark etc. In order to use these characters, escape sequence is used.

For example: \n is used for newline. The backslash ( \ ) causes "escape" from the normal way the characters are interpreted by the compiler.

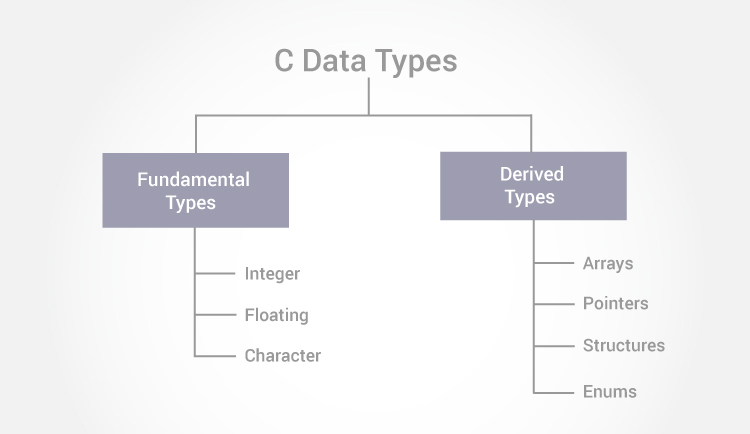
| Escape Sequences | |
| --- | --- |
| Escape Sequences | Character |
| \b | Backspace |
| \f | Form feed |
| \n | Newline |
| \r | Return |
| \t | Horizontal tab |
| \v | Vertical tab |
| \\ | Backslash |
| \' | Single quotation mark |
| \" | Double quotation mark |
| \? | Question mark |
| \0 | Null character |

### 6. Enumeration constants

Keyword enum is used to define enumeration types. For example:

enum color {yellow, green, black, white};

Here, color is a variable and yellow, green, black and white are the enumeration constants having value 0, 1, 2 and 3 respectively. For more information, visit page: [C Enumeration](https://www.programiz.com/c-programming/c-enumeration).



n C programming, variables or memory locations should be declared before it can be used. Similarly, a function also needs to be declared before use.

Data types simply refers to the type and size of data associated with [variables](https://www.programiz.com/c-programming/c-variables-constants) and [functions](https://www.programiz.com/c-programming/c-functions).

## Data types in C

1. Fundamental Data Types
   * [**Integer types**](https://www.programiz.com/terms/i/integer)
   * [**Floating type**](https://www.programiz.com/terms/f/floating-point)
   * [**Character type**](https://www.programiz.com/terms/c/character)
2. Derived Data Types
   * [**Arrays**](https://www.programiz.com/c-programming/c-arrays)
   * [**Pointers**](https://www.programiz.com/c-programming/c-pointers)
   * [**Structures**](https://www.programiz.com/c-programming/c-structures)
   * [**Enumeration**](https://www.programiz.com/c-programming/c-enumeration)

Int id;

Int id ,age;

Float accountBallance:

Double price;

float normalizationFactor = 22.442e2;

char test = 'h';

long double i;

### Volatile qualifiers

A variable should be declared volatile whenever its value can be changed by some external sources outside the program. Keyword volatile is used for creating volatile variables.

## xample #2: C Integer Output

#include <stdio.h>

int main()

{

int testInteger = 5;

printf("Number = %d", testInteger);

return 0;

}

**Output**

Number = 5

Inside the quotation of printf() function, there is a format string "%d"(for integer). If the format string matches the argument (testInteger in this case), it is displayed on the screen.

## Example #3: C Integer Input/Output

#include <stdio.h>

int main()

{

int testInteger;

printf("Enter an integer: ");

scanf("%d",&testInteger);

printf("Number = %d",testInteger);

return 0;

}

**Output**

Enter an integer: 4

Number = 4

## Example #3: C Floats Input/Output

#include <stdio.h>

int main()

{

float f;

printf("Enter a number: ");

// %f format string is used in case of floats

scanf("%f",&f);

printf("Value = %f", f);

return 0;

}

**Output**

Enter a number: 23.45

Value = 23.450000

## Example #4: C Character I/O

#include <stdio.h>

int main()

{

char chr;

printf("Enter a character: ");

scanf("%c",&chr);

printf("You entered %c.",chr);

return 0;

}

**Output**

Enter a character: g

You entered g.

## Example #5: C ASCII Code

#include <stdio.h>

int main()

{

char chr;

printf("Enter a character: ");

scanf("%c",&chr);

// When %c text format is used, character is displayed in case of character types

printf("You entered %c.\n",chr);

// When %d text format is used, integer is displayed in case of character types

printf("value of %c is %d.", chr, chr);

return 0;

}

**Output**

Enter a character: g

You entered g.

ASCII value of g is 103.

The ASCII value of character 'g' is 103. When, 'g' is entered,

## Example #6: C ASCII Code

#include <stdio.h>

int main()

{

int chr = 69;

printf("Character having ASCII value 69 is %c.",chr);

return 0;

}

**Output**

Character having ASCII value 69 is E.

C operator:

An operator is a symbol which operates on a value or a variable. For example: + is an operator to perform addition.

C programming has wide range of operators to perform various operations. For better understanding of operators, these operators can be classified as:

### Operators in C programming

### Arithmetic Operators(+,-,\*,/,%)

### Increment and Decrement (++,--)

### Assignment Operators(=)

### Relational Operators(==, < > !=)

### Logical Operators(&& || !)

### Conditional Operators Terinary(a ==1 ? a b)

### Bitwise Operators(&, |,^,<<,>>,)

### Special Operator:

### The sizeof operator

The sizeof is an unary operator which returns the size of data (constant, variables, array, structure etc).

### Example #6: sizeof Operator

#include <stdio.h>

int main()

{

int a, e[10];

float b;

double c;

char d;

printf("Size of int=%d bytes\n",sizeof(a));

printf("Size of float=%d bytes\n",sizeof(b));

printf("Size of double=%d bytes\n",sizeof(c));

printf("Size of char=%d byte\n",sizeof(d));

printf("Size of integer type array having 10 elements = %d bytes\n", sizeof(e));

return 0;

}

**Output**

Size of int = 4 bytes

Size of float = 4 bytes

Size of double = 8 bytes

Size of char = 1 byte

Size of integer type array having 10 elements = 40 bytes