

# C PROGRAMMING



## What is C

C is a programming language developed at AT & T's Bell Laboratories of USA in 1972. It was designed and written by a man named Dennis Ritchie. In the late seventies C began to replace the more familiar languages of that time like PL/I, ALGOL, etc. Possibly why C seems so popular is because it is reliable, simple and easy to use. An opinion that is often heard today is – "C has been already super coded by languages like C++, C# and Java".

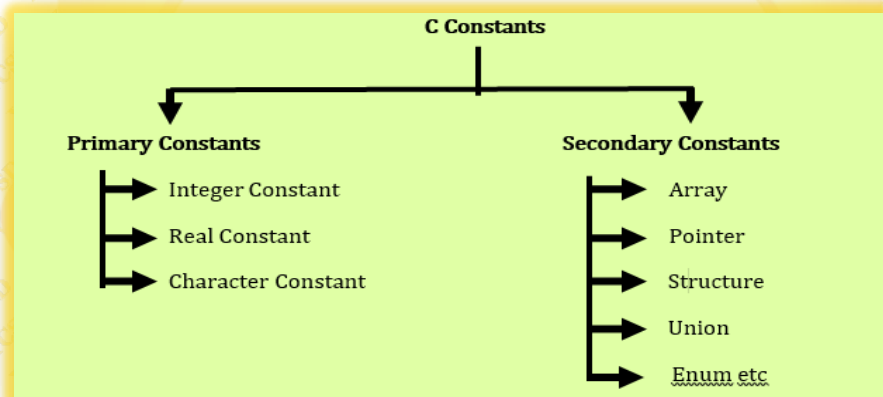
## The C Character Set

Alphabets	A, B, ....., Y, Z a, b, ....., y, z
Digits	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Special symbols	~ ' ! @ # % ^ & * ( ) _ - + =   \ { } [ ] ; : " ' < > , . ? /

## Types of C Constants

C constants can be divided into two major categories:

- ✓ Primary Constants
- ✓ Secondary Constants



## Rules for Constructing Integer Constants

- ✓ An integer constant must have at least one digit.
- ✓ It must not have a decimal point.
- ✓ It can be either positive or negative.
- ✓ If no sign precedes an integer constant it is assumed to be positive.
- ✓ No commas or blanks are allowed within an integer constant.
- ✓ The allowable range for integer constants is -32768 to 32767.

Ex.: 426, +782, -8000, -7605

## Rules for Constructing Real Constants

Real constants are often called Floating Point constants. The real constants could be written in two forms— Fractional form and Exponential form. Following rules must be observed while constructing real constants expressed in fractional form:

- ✓ A real constant must have at least one digit.
- ✓ It must have a decimal point.
- ✓ It could be either positive or negative.
- ✓ Default sign is positive.
- ✓ No commas or blanks are allowed within a real constant.

Ex.: +325.34, 426.0, -32.76 -48.5792



## Rules for Constructing Character Constants

- ✓ A character constant is a single alphabet, a single digit or a single special symbol enclosed within single inverted commas. Both the inverted commas should point to the left. For example, 'A' is a valid character constant whereas 'A' is not.
- ✓ The maximum length of a character constant can be 1 character. **Ex.: 'A', 'I', '5', '='**

## Types of C Variables

As we saw earlier, an entity that may vary during program execution is called a variable. Variable names are names given to locations in memory. These locations can contain integer, real or character constants. In any language, the types of variables that it can support depend on the types of constants that it can handle. This is because a particular type of variable can hold only the same type of constant. For example, an integer variable can hold only an integer constant, a real variable can hold only a real constant and a character variable can hold only a character constant.

## Rules for Constructing Variable Names

- ✓ A variable name is any combination of 1 to 31 alphabets, digits or underscores. Some compilers allow variable names whose length could be up to 247 characters. Still, it would be safer to stick to the rule of 31 characters. Do not create unnecessarily long variable names as it adds to your typing effort.
- ✓ The first character in the variable name must be an alphabet or underscore.
- ✓ No commas or blanks are allowed within a variable name.
- ✓ No special symbol other than an underscore (as in `gross_sal`) can be used in a variable name.  
**Ex.: `si_int` or `m_hra` or `pop_e_89`**

## C Keywords

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

## The First C Program

Before we begin with our first C program do remember the following rules that are applicable to all C programs:

- ✓ Each instruction in a C program is written as a separate statement. Therefore, a complete C program would comprise of a series of statements.
- ✓ The statements in a program must appear in the same order in which we wish them to be executed; unless of course the logic of the problem demands a deliberate 'jump' or transfer of control to a statement, which is out of sequence.
- ✓ Blank spaces may be inserted between two words to improve the readability of the statement. However, no blank spaces are allowed within a variable, constant or keyword.
- ✓ All statements are entered in small case letters.
- ✓ C has no specific rules for the position at which a statement is to be written. That's why it is often called a free-form language.
- ✓ Every C statement must end with a ;. Thus ; acts as a statement terminator.

## Data types Format Specifies for I/O

Type	Size (bytes)	Format Specifier
int	at least 2, usually 4	%d
char	1	%c
float	4	%f
double	8	%lf
short int	2 usually	%hd
unsigned int	at least 2, usually 4	%u
long int	at least 4, usually 8	%li
long long int	at least 8	%lli
unsigned long int	at least 4	%lu
unsigned long long int	at least 8	%llu
signed char	1	%c
unsigned char	1	%c
long double	at least 10, usually 12 or 16	%lf

### Int

Integers are whole numbers that can have both zero, positive and negative values but no decimal values.

For example, 0, -5, 10

**We can use int for declaring an integer variable.**

- ✓ int id;

### float and double

Float and double are used to hold real numbers.

- ✓ float salary;
- ✓ double price;

### char

Keyword char is used for declaring character type variables.

For example, char test = 'h';

### void

Void is an incomplete type. It means "nothing" or "no type". You can think of void as absent.

**For example**, if a function is not returning anything, its return type should be void.

### short and long

**If you need to use a large number, you can use a type specifier long. Here's how:**

long a;

long long b;

long double c;

Here variables a and b can store integer values. And, c can store a floating-point number.

If you are sure, only a small integer ([-32,767, +32,767] range) will be used, you can use short.

short d;

## C Input Output (I/O)

**scanf()** function to take input from the user, and

**printf()** function to display output to the user.



## Let's Start Programming.....

### Print your name

```
#include<stdio.h>
#include<conio.h>
main ()
{
printf("Rahul Chatterjee ");
getch();
}
```

#### Output:

Rahul Chatterjee

### Print your name, address, and ph no in separate line

```
#include<stdio.h>
#include<conio.h>
main ()
{
printf("Rahul Chatterjee\n ");
printf("Kolkata\n ");
printf("9088654789 ");
getch();
}
```

#### Output:

Rahul Chatterjee  
Kolkata  
9088654789

### Program to find sum of two numbers

```
#include<stdio.h>
#include<conio.h>
main ()
{
int a,b,s; a=10; b=20;
s=a+b;
printf("Sum=%d",s);
getch();
}
```

#### Output:

Sum=30

### Program to find sum of two numbers using scanf()

```
#include<stdio.h>
#include<conio.h>
main()
{
int a,b,s;
printf("Enter two no: ");
scanf("%d%d",&a,&b);
s=a+b;
printf("Sum=%d",s);
getch();
}
```

#### Output:

Enter two no:  
5 6  
Sum=11

### Write a C program enter two numbers and performs all arithmetic [ +,-,\*,and / ]

```
#include<stdio.h>
#include<conio.h>
main()
{
int a,b,sum,sub,multi,div;
printf ("Enter value of a:");
scanf("%d",&a);
printf ("Enter value of b:");
scanf("%d",&b);
sum=a+b;
sub=a-b;
multi=a*b;
div=a/b;
printf("Sum of a and b is =%d\n",sum);
printf("Sub of a and b is =%d\n",sub);
printf("Multi of a and b is =%d\n",multi);
printf("Div of a and b is =%d",div);
getch();
}
```

#### Output:

Enter value of a: 20  
Enter value of b: 10  
Sum of a and b is = 30  
Sub of a and b is = 10  
Multi of a and b is = 200  
Div of a and b is = 2

# Welcome In YCSD

For Access Entire Study Materials

**For Student**

Please Ensure Your Enrollment Verification

## ENROLLMENT VERIFICATION

Student Registration No.

Date Of Birth

Check Now

**For Franchise**

Please Login Via Franchise Id & Password

## CENTER LOGIN

User Name

Password

Login

Thank You for Visiting Us

YCSD

 Youth Computer Skill Development

 YCSD

 [www.ycsd.in](http://www.ycsd.in)

 [info@ycsd.in](mailto:info@ycsd.in)



[www.ycsd.in](http://www.ycsd.in)