

\* Introduction:-  
Background

C language is a structured, high level language, standard and most powerful programming language. It was developed by 'Denis Ritchie' in 1972 at Bell Laboratory in the USA (Maury hill). It is also known as function oriented programming. It is the advance form of 'B' language, which was developed by Ken Thompson. It was designed/ developed to re-designed the UNIX operating system.

There are some example of uses of C language:

- i) Designing of data base system
- ii) Designing of graphics package
- iii) Word Processor
- iv) Spreadsheet
- v) OS development
- vi) Compilers and assemblers, etc.

\* Variable:-

→ Variable is a name which stores a value and value of variable may be change according to requirement.

For variable, we need to memorize some points.

- i) The first character of name of variable is always alphabet or underscore.
- ii) Space is not allowed b/w the name of variable.
- iii) The <sup>max</sup> length of variable may be 31 characters.
- iv) Only one special character underscore (-) is allowed with the name of variable.

\* There are two types of variable :-

- i) Local variable :- It's by default value is garbage.
- ii) Global variable :- It's by default value is zero.

Ex:-

```
#include <stdio.h>
#include <conio.h>
int x; // Global variable
Void add()
{
```

```
    int y; // local variable
```

```
    }
```

```
Void main()
{
```

```
    int z; // local variable
```

```
    }
```



\* Constant:— It is fixed number or value or characters, which value can't be change.

→ There are four types of constant:

- i) integer constant
- ii) float constant
- iii) character constant
- iv) string constant

i) Integer Constant:— All integer numbers which value is fixed, is known as integer constant. Ex:- 1, 2, 3, ... etc.

ii) Float Constant:— All float numbers is known as float constant because its value is fixed. Ex:- 6.3, -3.9, 5.0 etc.

iii) Character Constant:— The character which is within the single inverted comma is known as character constant.  
Ex:- 'a', 'b', 'c', 'i', 'g' etc.

iv) String Constant:— The group of character written within the double inverted comma is known as string constant.  
Ex:- "Enter any two numbers" etc.



## \* Keywords :-

Keywords is also known as reserve words. The words which meaning is already defined in the C language library is known as "C language keywords" or "keywords".

There are 32 keywords in C language.

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

## \* Datatype :-

It is a way which define how many space taken by any variable in the computer memory.

There are diff. types of datatype:-

i) Basic datatype:-

There are 3 types of basic datatype.



datatype	format	space taken in RAM	Range
int	%d	2 byte	$-2^{15}$ to $2^{15} - 1$ (-32768 to 32767) (0 to 65535)
float	%f	4 byte	$-2^{31}$ to $2^{31} - 1$
char	%c	1 byte	$-2^7$ to $2^7 - 1$ (-128 to 127) (0 to 255)

\* Identifier :-  
Identifier is a name, which is used to identify the variables, constants and functions etc.

→ The identifier may be of <sup>max.</sup> 31 characters.  
→ The first letter of any identifier will be alphabet or underscore ; can't be numbers.  
Ex:- x, sum, add(), c2(), a1 etc.

Here, x, sum, a1 is a variable and add(), c2() are the functions.

### \* Operator :-

→ Operator is a symbol, which is used to perform some operations in the computer system.

→ There are 8 types of operator :-

- (i) Arithmetic Operator
- (ii) Relational Operator
- (iii) Logical Operator
- (iv) Increment / Decrement Operator
- (v) Assignment Operator
- (vi) Conditional Operator / Ternary Operator
- (vii) Bitwise Operator
- (viii) Special Operator

Shot on Y12

Vivo AI camera



(i) **Arithmatic Operator** :- The operator which is used to perform the arithmathical operation is known as arithmetic operator.

→ There are five types of arithmatic operator.  
Ex:- +, -, \*, /, %.

Arithematic Operator	Name of arithematic operator	Example	Result
+	plus	5+2	7
-	Minus	5-2	3
*	Multiplication	5*2	10
/	division	5/2	2
%	Mod	5%2	1

(ii) **Relational Operator** :- The operator which is used to perform the relation between two numbers or variables is known as relational operator.

→ There are six types of relational operator.  
Ex:- <, >, <=, >=, ==, !=.

→ It always provide the result in the form of true or false.

Relational Operator	Name of relational operation	Example	Result
<	Less than	$5 < 2$	F
>	Greater than	$5 > 2$	T
$\leq$	Greater than equal to	$5 \leq 2$	F
$\geq$	Less than equal to	$5 \geq 2$	T
$=$	Equal to	$5 = 2$	F
$\neq$	Not equal to	$5 \neq 2$	T

### (iii) Logical Operator:-

- Logical operator is a type of operator which is used to perform the logical operations in the computer system.
- It shows the result in the form of true or false.
- It is used with arithmatical or relational operations.
- There are three types of logical operators:-

$\&&$ ,  $||$ , !



Logical Operator	Name of logical operator	Example	Result
&&	logical AND	$(5>2) \&\& (3>4)$ T F	F
	logical OR	$(5>2)    (3>4)$ T F	T
!	logical NOT	!( $5>2$ ) T	F

#### (iv) Increment / Decrement Operator :-

$++$  Increment Operator

→ Increment operator is used to increase the value of variable by 1. It is denoted by  $++$ .

Syntax:-

$(\text{variable})++$   
 $++(\text{variable})$

Expt-

$$x++ \Rightarrow x = x + 1$$

$$++x \Rightarrow x = x + 1$$

Ex:-

$$\text{Let, } x = 6$$

$$x++ \Rightarrow x = x + 1 = 6 + 1 = 7$$

$$\text{or, } ++x \Rightarrow x = x + 1 = 6 + 1 = 7$$

$--$  Decrement Operator

→ If is used to decrease the value of variable by 1. It is denoted by  $--$ .

Syntax:-

$(\text{variable})--$   
 $--(\text{variable})$

Expt -

$$x-- \Rightarrow x = x - 1$$

$$--x \Rightarrow x = x - 1$$

Ex:-

$$\text{Let, } x = 6$$

$$x-- \Rightarrow x = x - 1 = 6 - 1 = 5$$

$$--x \Rightarrow x = x - 1 = 6 - 1 = 5$$

## (V) Assignment Operator:-

- Assignment operator is that operator in which is used to assign the value of RHS in the variable of L.H.S.
- It is denoted by equal (i.e.,  $=$ ).
- Ex :-  $x = 5$

→ There are five additional assignment operators

- i)  $+ =$
- ii)  $- =$
- iii)  $* =$
- iv)  $/ =$
- v)  $\% =$

Operator	Name	Expression with example (Let $x=3$ )	Result
$+ =$	plus equal to	$x + = 5$ $x = x + 5$	8
$- =$	minus "	$x - = 5$ $x = x - 5$	-2
$* =$	multiple "	$x * = 5$ $x = x * 5$	15
$/ =$	division "	$x / = 5$ $x = x / 5$	0
$\% =$	mod "	$x \% = 5$ $x = x \% 5$	3

## (Vi) Conditional Operator / Ternary Operator:-

- Conditional Operator is a type of operator which is used to represent the conditional statement & more than one line in



a single line  
Syntax :-

[ variable = (condition)? true value : false value ; ]

Ex:-

$$\left. \begin{array}{l} \text{if } (a > b) \\ \quad \text{big} = a \\ \text{else} \\ \quad \text{big} = b \end{array} \right\} \Rightarrow \text{big} = (a > b)? a : b$$

(Vii) Bit-wise operator :-

→ Bit-wise Operator is a type of operator which works on bit value.

→ This operator is used by the system for various works, this is not used by us.

→ There are six types of operator :-

⇒ Right shift

⇒ Left shift

& bitwise logical AND

| bitwise logical OR

^ bitwise logical XOR

~ Complement / logical NOT

Ex:-

Right shift:-

Let,  $a = 5$

~~a~~  $a \gg 2$

0000001101

000000001  $\Rightarrow 1$

Left shift:-

Let,  $a = 5$

$a \ll 2$

101001011011

101001101100  $\Rightarrow 20$

### (viii) \* Special Operators:-

→ It is a type of operator which is used for special purpose / work.

There are diff. type of special operators:-

i) `sizeof()` operator:-

It is a type of special operator which is used to show the size of any variable or datatype.

Ex:-

```
int x;
int y;
int z;
```

`sizeof (int)` → 2

`sizeof (float)` → 4

`sizeof (char)` → 1

`sizeof (d)` → 2

`sizeof (y)` → 4

`sizeof (z)` → 1

(iii) Comma operator :-

It is a type of special operator which is used to show the similar type of expression in a single line using comma operator.

Ex:-

`temp = a ;`

`a = b ;` ⇒ `temp = a, a = b, b = temp`

`b = temp ;`

\* Structure of C-language file

`#include <stdio.h>`

Preprocessor

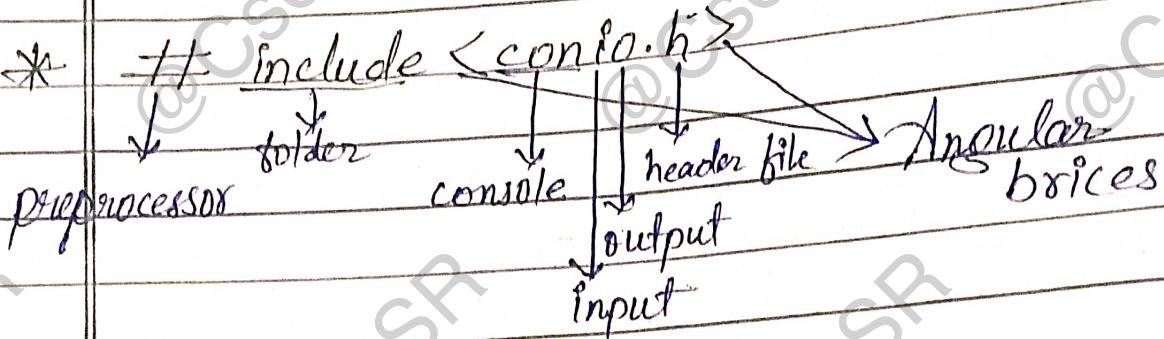
↑  
folder

header file  
output  
input

Angular  
braces



`printf()` → `stdio`  
`scanf()` → `stdi`



`clrscr()` → `conio`  
`getch()` → `conio`

\* `printf()` function :—

`printf()` function is a standard output function. Using this, we can print message, value and both in combined way.

Syntax :—

(i) `printf("msg");`

Ex:-

① `printf("enter any no.");`

O/P :—

enter any no.

② `printf ("format", variable);`

Ex:- `printf "x = %d";`

Vivo AI camera



printf ("%d", x);

O/P:-

5

③ printf ("msg = format", variable);

Ex:- int x=5

printf ("%d", x);

O/P:-

x=5

→ Token :-

Token is the smallest individual element of 'C' program.

→ There are six types of tokens:-

- i) Keywords
- ii) Identifier
- iii) Operator
- iv) Constants
- v) String
- vi) Separator

printf ("%d", x);

O/P:-

5

③ printf ("msg = format", variable);

Ex:- int x=5

printf ("x = %d", x);

O/P:-

x = 5

\* Token:-

Token is the smallest individual element of 'C' program.

There are six types of tokens:-

- i) Keywords
- ii) Identifier
- iii) Operator
- iv) Constants
- v) String
- vi) Separator

Q. Write a program to print your name.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    clrscr();
    printf("Surabhi Pranjal");
    getch();
}
```

Q. WAP to print your five friends name.

/\* to print my five friends name \*/

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
void main()
```

```
{
```

```
clrscr();
```

```
printf("Muskan \n Rashni \n Juhi\n
       \n Komal \n Bhavya");
```

```
getch();
```

```
}
```

Q. WAP to print

\*  
\*\*  
\*\*\*  
\*\*\*\*  
\*\*\*\*\*

```

#include <stdio.h>
#include <conio.h>
void main()
{
  clrscr();
  printf ("* \n ** \n *** \n **** \n ***** \n *****");
  getch();
}
  
```

Q) WAP to print

\*  
 \* \* \*  
 \* \* \* \*  
 \* \* \* \* \*  
 \* \* \* \* \* \*

```

#include <stdio.h>
#include <conio.h>
void main()
{
  clrscr();
  printf ("* \n ** \n *** \n **** \n ***** \n *****");
  getch();
}
  
```

\*—\*—\*—\*—\*—\*—\*—\*

Note: \* Alt + F5 = result

\* Alt + F9 = compile

\* Ctrl + F9 = run

\* \n is used to go to next line.

Q. WAP to print 1 2 3  
4 5 6

```
#include <stdio.h>
#include <conio.h>
void main()
{
    clrscr();
    printf("1\n2\n3\n4\n5\n6");
    getch();
}
```

Q. WAP to print \* \* \* \*  
\* \* \* \*

```
#include <stdio.h>
#include <conio.h>
void main()
{
    clrscr();
    printf(". * \n . * . * \n . * * * \n * * * *");
    getch();
}
```

Note:-

- \* Comment line is used to write something on the 'c' program editor except program. It is not displayed on the output screen.



Shot on Y12

Vivo AI camera

2022.04.16 21:40

## Files used in C-program :-

Ex :-

Header file	-	stdio.h
Source code file	-	xam.c
Object file	-	xam.obj
Executable file	-	xam.exe
Backup file	-	xam.bac

### Header file :-

Header file is a type of 'C' programming file. In header file meaning of all functions defined, which is used in 'C' programming like; printf() and scanf() function defined in "stdio" header file, clrscr() and getch() and other functions like this defined in "conio" header file.

There are different types of header file :-

stdio.h

conio.h

math.h

string.h

doc.h etc.

### Source code file :-

It is also a type of 'C' programming file. In 'C' programming language whenever



We write the program on editor of 'C' language, we save it with a name with extension code ".c". This file is known as source code file. It always open with 'C' language editor.

### iii) Object file:-

Whenever source code file is compiled an object file generated. Its extension code is ".obj". This file is in the form of binary.

### iv) Executable file:-

Whenever we execute (or run) the source code file, executable file generated. It execute and provide us the output. Its extension code is ".exe".

### v) Backup file:-

It is also a type of file, which is generated by the computer system itself to take the backup of that program. Its extension code is ".bac".

A

## Type conversion and Type casting:-

Type conversion or type casting is a way



Shot on Y12

Vivo AI camera



DATE: \_\_\_/\_\_\_/\_\_\_

PAGE: \_\_\_

to convert the variable of a datatype into another datatype at the particular time period or particular operation.

Ex:-

```
int x, z;
```

```
float y;
```

```
y = (float) x/z;
```

variable = (datatype to ) variable / variable or numbers  
                          be converted

\* scanf() function:-

scanf() function is a standard input function. It is used to read the value of variable of diff. datatypes.

Syntax:-

```
scanf ("format", & variable);
```

Ex:-

```
int x, y, z;
```

```
float w;
```

i) 

```
scanf ("%d", & x);
```

ii) 

```
scanf ("%d %d %d", & x, & y, & z);
```

iii) 

```
scanf ("%f", & w);
```

iv) 

```
scanf ("%d %d %f %d", & x, & y, & w, & z);
```

Q. WAP to find the addition of two numbers.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int x, y, sum ;
    clrscr();
    printf ("In enter the value of x and y");
    scanf ("%d %d", &x, &y);
    sum = x + y ;
    printf ("In sum = %.d", sum);
    getch();
}
```

Q. WAP to find simple interest.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int p, r, t, SI ;
    clrscr();
    printf ("In enter the value of p, r, &t");
    scanf ("%d %d %d", &p, &r, &t);
    SI = p * r * t / 100.0 ;
    printf ("In SI = %.d, SI");
    getch();
}
```



8. WAP to find area of circle.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int r;
    float area;
    clrscr();
    printf("In enter the value of r");
    scanf("%d", &r);
    area = 3.14 * r * r;
    printf("In Area = %.f", area);
    getch();
}
```

B WAP to find area of triangle using heron's formula

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
void main()
{
    float a, b, c, area, s;
    clrscr();
    printf("In enter the value of a, b, c");
    scanf("%f, %f, %f", &a, &b, &c);
    s = (a+b+c)/2;
    area = pow((s*(s-a)*(s-b)*(s-c)), 0.5);
    printf("In Area = %.f", area);
    getch();
}
```

- ④ WAP to find addition and multiplication of three no's.

```
#include <stdio.h>
#include <conio.h>
void main()
{
```

```
float a, b, c, mul, add;  
clrscr();
```

```
printf ("Please enter the value of a, b, c");
```

```
scanf ("%f,%f,%f", &a, &b, &c);
```

$$\text{add} = a + b + c;$$

$$\text{mul} = a * b * c ;$$

```
printf("In addition = %f", add);
```

```
printf("In multiplication=%f", mul);
```

getch();

- ⑤ WAP to find compound interest.

~~#include <stdio.h>~~

```
#include <conio.h>
```

```
#include <math.h>
void main()
```

1

float p, t, CI;

~~clsscr();~~

```
printf("In enter the value of n is +");
```

```
scanf ("%f,%f,%f,&p,&r,&t);
```

~~EE~~ - ~~Project 1000~~

```

CT = p * pow((1+r/100), t) - p;
printf ("\n C.I = %.f ", CI);
getch();
}

```

⑥ WAP to find the average of 4 no's.

```
#include < stdio.h >
```

```
#include < conio.h >
```

```
void main()
```

```
{
```

```
float a, b, c, d, avg;
```

```
clrscr();
```

```
printf ("In enter the value of a, b, c, d");
```

```
scanf ("%f, %f, %f, %f", &a, &b, &c, &d);
```

```
avg = (a+b+c+d)/4;
```

```
printf ("In Avg = %.f", avg);
```

```
getch();
```

```
}
```

## ~~Characteristics of C~~

- i) It is a general purpose programming language.
- ii) It is a structured programming language.
- iii) Flexibility towards system programming.
- iv) Flexibility towards application programming.
- v) It has small instruction set.
- vi) Ability of large number of operand and inbuilt function.
- vii) Ability of computer for all types.