

C - Programming -

- * C is a procedural oriented language.
- * It is a structured programming language.
- * It also contains additional features that allow it to be used at a lower level.
- * C is machine independent.
- Highly portable language.

Characteristics of C language :-

- Simple and clear language.
- Portable
- Case sensitive.
- Easily available and quick
- Modular.
- Easy error detection.
- Structured approach.
- Lesser keywords.
- Compact code.
- Availability of recursive function.
- Availability of memory management function.
- System programming.

Uses :-

Writing software

Translator

Assembler.

Make operating system.

Network Software

Graphics programming.

Drawbacks -

Not provide Data Security

- Does not support reusability of source code.
- Procedure oriented language.
- Doesn't support Object Oriented Programming.
- C does not have built-in collection library.

Computer Programming :-

→ Process of writing, testing and maintaining the source code of computer programs.

* Program Development :-

Steps -

- 1.) Problem definition and analysis.
- 2.) Program design.
- 3.) Coding.
- 4.) Compilation
- 5.) Debugging and testing
- 6.) Documentation.
- 7.) Implementation and Maintenance.

General and Basic structure of C program-

- 1) Documentation
 - 2) Link section -
 - 3) Global declaration
 - 4) Definition
 - 5) Function prototype
 - 6.) Main function
- extension.
- Declaration
 - Definition
 - Execution.

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

preprocessor directive :

<conio.h>

console output screen

Independent

Function - It is a small piece of code which is used
 Built in. perform a specific type of task.
 User defined.

Ques Write a C Basic program to print a msg.

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

{

```
/* First simple C basic program to print a
msg */
```

```
printf ("Hello, How are you ?");
```

```
getch ();
```

Termination.

}

Output -

Hello, How are you?

Print function

Date - 30/09/21

ASCII codes

American standard code for information interchange.

Run turbo C

Types of Errors:-

Compile time errors / Syntax error - Error which are generated at the time of compilation due to incorrect syntax. These errors are displayed after the compilation and can be removed easily as per the suggestion of compiler dialog box.

Run time errors:-

The errors which are generated at run time is called Runtime. These occur whenever programmer try to violate the rule of programming language.

Logical Errors:-

Error which are generated in the logic of the program and give the incorrect & unexpected errors are called

Warning - It is also an abnormal condition whenever it is occurred execution of program never be stopped.

Character set of C language -

Set of character which can be used to represent information in a language.

Tokens - Lexical Unit of program -

smallest individual unit of program that cannot be broken down further unless it retain its meaning.
It is the \Rightarrow

Constant :- Quantity that does not change with the execution of program.

Int

float, real.

Ch, Str.

Floating - Constant with decimal point.

Fractional

Exponential. - mantissa, exponent.

Character Constant -

Escape Sequence

String Constant -

Variable - Name given to memory location where different constants are stored. It may vary (change) during the execution of program.

Identifiers - Name given to various elements of program such as variable, arrays etc.

Keyword - Reserved words, already being defined. 32.

Homogeneous data elements

Arrays - data type array name [size];
 eg - int bca[60];

Structures - Type of secondary data type in which different type of data type can be merged in a single variable.

struct structure name

{

 1st data / variable
 2nd data / ..

}

Union - (same as structure).

- Use for space utilisation.
- memory taken by union = length of variable that has the max. size.

Pointers -

Derived data type and special type of variable which is used to store the memory of another variable

e.g -

int a,*b;

a = 5;

b = &a;

User defined :-

typedef - which is used to provide short & meaningful name to an existing data type.

typedef previous name new name;

Enum:- (Enumerated data type)

Used to create our own data type along with some predefined values.

→ enum class - Data
{
 type.

};

enum class S1, S2;

Symbolic constants - It is a name that substitutes a constant value for particular name which cannot change.

Character may represent numeric, character or string. When the program is compiled, each occurrence of symbolic constant replaced by its corresponding character sequence.

define.

Expression -

→ Legal combination of symbol that represents a value. Every expression consist of at least one operand and can have one or more operators.

eg - $x + 5$
operator
variable.

C statements / instruction -

→ These are written out with some rules and regulation

* If there were more than one operators with same category then associativity is method that determine the which operators can be executed. (L to R)
(R to L)

PAGE NO.	
DATE :	/ /

OPERATORS AND EXPRESSION -

Operations :-

* Symbol or letter used to indicate specific operation on variable in program. Operands may include constant, variable.

Type :-

- | | | |
|---------|-------------------|------------------------|
| Type :- | <u>Precedence</u> | <u>* Associativity</u> |
| | *, /, % | +,- |
1. Arithmetic Operators - Various arithmetic calculation eg, +, -, *, /, %
 2. Relational -
 3. Logical -

$$\begin{aligned}y &= 5 - 8/2 + 9 * 4 + 8/4 - 7 + 1 \\&= 5 - 4 + 36 + 2 - 7 + 1 \\&= 1 + 38 - 6 \\&= 39 - 6 \\&= 33.\end{aligned}$$

Type Conversion -

In some situation we may declare data type we need result some other data type then the process of conversion of one data type to other is called type conversion.

Two ways -

Automatic type conversion.

Type casting.

Type casting - Process of converting one data type into other as per the requirement of the user.

data-type (expression)

(int [x+y+z])/a.

e.g -

{ int a=100, b=40;

float c

c=a/b; — 2.0

c=(float)a/b — 2.5.

}

Relational Operators -

value of the

- Operator used to compare the operands.
 - Result of such operation is always logical either true [1] or false [0]
 - Decision making statements .
- >, >=, <, ≤ , == , !=.

Logical operators -

Used to connect relational or logical and there result is always 1 True or 0 False.

1. NOT (Negation) - It reverse the value of operand. (unary)
2. OR(||) (disjunction) - Give true result if any one are true.
3. AND (.) (conjunction) - Returns true only if both input are true
(&&) (Binary)

Assignment Operations -

Used to assign the value of r.H.S operand to L.H.S operand

$$\text{sum} = a + b$$

Shorthand Operations -

These were rewritten with the help of = operator and they are use to reduce the size of code.

Identifier Arith. Operator = Expression.

→ These operators that act upon single operand to produce new value.

'unary minus (-)

Increment (Increase the value by 1).

↓ Pre- $a++$

Post- $a++$

(first use then point).
(increase)

→ Decrement (Decrease the value by 1)

→ Address of Operator (&)

* Use it to get the address of any operand (variable) from the memory location.

$y = &x;$

→ Value at address operator (Indirection operator)

* It is used to make a normal variable into pointer variable
convert.

→ Size of Operator:- Return the size.

→ Conditional operator (Ternary).

* Which are used as a replacement of if, else statement.

exp1?exp2:exp3

$z = (x > y) ? x : y;$

Bitwise Operator :-

→ These operate at bit level and allows the user to manipulate individual bits.

→ These operators are used at low level i.e. Machine level language.

Left shift - shift the bit to left side

2 | 1 0
2 | 0 0

Right shift - " " " right side

2 | 0 0
2 | 0 0

One's complement - Change each 1 to 0

2 | 0 0
2 | 0 0

2 | 0 0
2 | 0 0

Comma Operator :-

0 0 1 0 0

Used to group related statements or expression. 1 0 0 0 0.

Library function -

Control Statement - Used to control the flow of instruction within a program. These instructions are executed sequentially, but by using these statements we can change the order of execution.

1. Decision
2. Case control
2. Selection
3. Looping
3. Structure
4. Jumping

Decision :- Special kind of instruction which are used for particular result based on some predefined condition.
if, else, if-else, nested if-else, else-if ladder, case control structure, conditional operator.

if (condition)

[Statement block]

if (condition)

[Statement 1st]

else

[Statement 2nd block]

Looping - set of statement should be executed more than one time.

while

for

do-while;

Counter - Special type of variable which is used to count the execution of the loop. It is incremented or decremented by a fix amount and stop the execution as per the condition invoke by user.

While loop :-

True-Test loop -

while (condition)

[statement block]

For statement :- when we don't known how many time a loop is to be executed.

for

initialization / expression / (increment & decrement) / (statement block).

Some key points regarding for loops -

Mostly used with switch statement after the usage of break statement for loops - next statement is skipped and control goes out of the block.

Jumping statements :- Used to transfer the control from one part of program to another.

Break : Transfer the control out of the coming closing parenthesis

Continue :

Go to

Array :- Collection of homogenous data elements of a specific data type that have been in single name, which are stored in contiguous memory location.

data type $s[size]$

variable name of array

Two dimensional -

int $ra[5][6]$

rows columns.

Visit and represent the array - Transverse.

Multidimensional - Two or more array or subscript.

$a[2][3][2]$

matrix |

Row Columns.
 in all

array of character - String or literal.

null string /0.

- o during its initialization its data type written as char whenever we declare string then size of string always greater than one of its size. last character is null string which generate space.

V V
V V
X

Input Output functions :-

Standard Library function :-

strlen [length]

strcat [append the string]

strcpy [copy string]

strcmp [compare string]

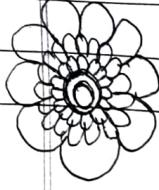
strrev [reverse the string]

strupr [uppercase]

strlwr [lowercase]

strcmp () return value less than 0 if $s_1 < s_2$, zero if
 $s_1 = s_2$. It returns ~~then~~ value more than 0

FUNCTION :-



DYNAMIC MEMORY ALLOCATION :-

Is achieve in C programming by using library function. These functions are used to allocate the memory to variables at run time and after the execution of program allocated memory block will be destroyed at run time. Hence this technique is very important for memory optimization.

~~V Imp~~

Aangling pointer :- It is a pointer that points to nowhere.

Four type of function which are used for dynamic memory allocation :-

- 1) `malloc()`
- 2) `calloc()`
- 3) `free()`
- 4) `realloc()`

`malloc()`

Reserve block of memory of the specified size and return a pointer specifies data type.

$\text{ptr} = (\text{data-type}^*) \text{malloc}(\text{No.-of-elements} * \text{size of}(\text{data-type}))$;

`Data-type*` :-

→ Use for data conversion or type casting.

Ex :-

`int* p;`

`p = (int*)malloc(5 * size of (int));`

→ Headerfile :- `stdlib.h`.

→ 10 memory bytes.

calloc():-

- It allocates space for an array of element and initialize them to zero then return a pointer to the memory.
- $\text{ptr} = (\text{int}^*) \text{calloc}(5, \text{size of } (\text{int})) ;$

free():-

- Deallocates or free a block of memory that was previously allocated. Takes a pointer to the block of memory as its argument and after its execution that block is no available.

→ Optimized programming

→ Syntax :-

$\text{int}^* p ;$

$\text{free}(p) ;$

Realloc():-

- Used to modify the size of previously allocated space
- Ex - $\text{p} = (\text{int}^*) \text{malloc}(5) ;$
If requirement change, it require 10 bytes.
 $\text{p} = (\text{int}^*) \text{realloc}(\text{p}, 10)$