

Topics	Page No
- Introduction to Programming:	2
- Concepts of Algorithm and Flowcharts	3
- Problem solving examples using algorithm and flowchart	3
- Types of Programming languages	5
- Characteristics of higher level language	5
- Compiler and Interpreter	5
- Overview of C: Introduction, Importance of C	5
- Sample C programs, Basic structure of C programs	7
- Programming style, executing of C program	7
- Constants, Variables and data Types	10
- Introduction, Character Set, C tokens	10
- Keywords and Identifiers, Constants	10
- Variables, Data types	12
- Declaration of Variables, Defining symbolic constants	14

❖ What is Computer ?

- A programmable machine.
- It responds to a specific set of instruction in a well-defined manner.
- It can execute prerecorded list of instruction.

❖ Part of computer:

- Software: It is a collection of program.
- Program: It is a collection of instruction.

❖ Instruction

- is command given by the user to the computer for performing special task.

❖ What is Language?

- Language is a communication tool using this two and more person can exchange their idea.
- Ex: Hindi, Gujarati, English etc.

❖ What is Computer Language?

- English is a International language and scientist develop computer language like English level language.
- It is not exactly English language but nearby English.
- Computer language have Own set of rules and alphabets.

❖ Types of Computer language:

- C, C++, VB, ASP.NET, JAVA, PHP etc.

❖ Concept of Algorithm and Flowcharts:

- Logic: Idea behind the solution of any problem is call the logic.
- Logic is differed from person to person. It is a sequence of stages for the solution of the problem.

❖ How to develop Logic?

- First of all understand problem definition.
- Then find out the actual process problem of solving.
- Than collect the necessary inputs for problem solving.
- Start the process for the problem solving.
- Generate the output which needed.

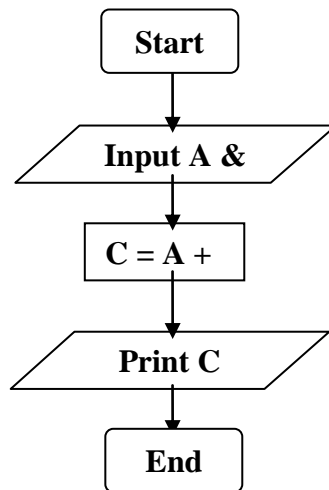
❖ What is Algorithm?

- Representation of the logic in the simple English statement is called Algorithm.
- Algorithm is a solution to a problem written in a step by step manner.

Example: Step: 1 Start
Step: 2 Input A,B
Step: 3 $C = A + B$
Step: 4 Print C
Step: 5 End

❖ What is Flowchart?

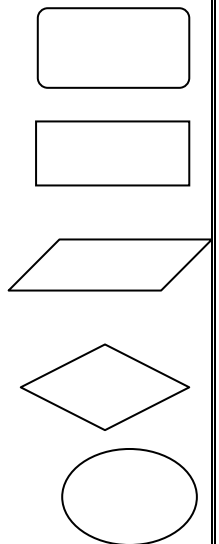
- Graphical or Symbolic representation of logic is calling the flowchart.
- Graphical or Symbolic representation of logic is calling the flowchart.



❖ How to draw flowchart?

We are use following Symbols to draw flowcharts.

- Start or End of the Program.
- Computational steps or processing function.
- Input out or output operation.
- Decision making and branching.
- Joining of two parts of program

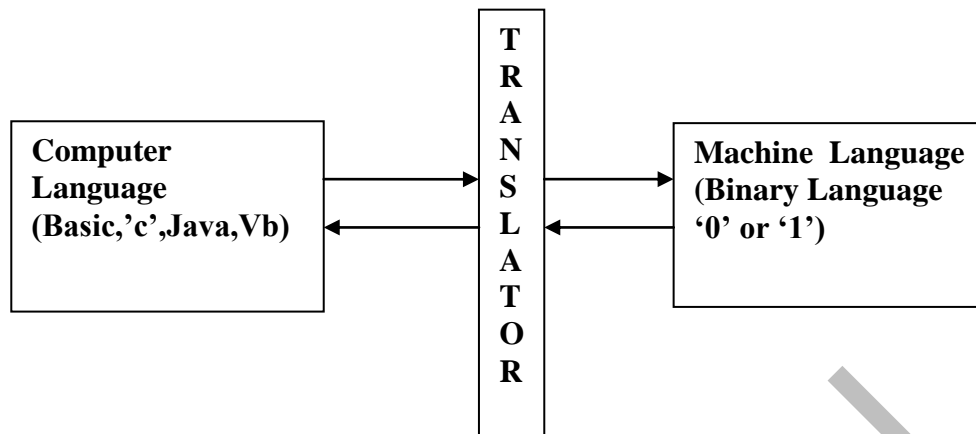


❖ What is Translator?

- Translator is a one type of software program that convert machinery level language in to computer language and also convert computer level language in to machine level language.
- Computer cannot understand any type of language because it is electronics machine it understand only and only machinery level language. (Binary language '0' or '1')

•





❖ There are two types translator are available:

- **Compiler:** It is compile whole program and find out the error. When solve all the error then convert into machine level language.
- **Interpreter:** Interpreter checks the error line by line and converts into machine level language line by line.
- Each and every Computer language has its own translator.

❖ Introduction of 'C' Language:

- 'C' is a powerful, flexible, portable and structured language.
- ANSI: American National Standard Institute.

❖ History of c language:

- It is a Programming language. It is a structured, high level machine Independent language.
- It allows software developers to develop programs without worrying about the hardware platforms where they will be implemented.
- It is most widely used general purpose language today.
- All popular computer languages are dynamic nature.
- They continue to improve their power and scope by new feature and 'C' is no exception.

- It was approved ISO (International standard organization)

1960	ALGOL	International group
1967	BCPL	Martin Richards
1970	B	Ken Thomson
1972	Traditional	Dennis Ritchie
1978	K&R C	Kernighan and Richie
1989	ANSI C	ANSI Committee
1990	ANSI/ISO C	ISO Committee

❖ Importance of 'C':

- Programs Written on c are efficient and fast. It has variety data types and powerful operators.
- It is highly portable.
- It is structured programming.

❖ Characteristics of 'C' Language:

- It has rich set of built in function.
- Operators can be used to write any programs
- Programs are efficient and fast.
- 'C' is suitable for making system software and business Packages.
- There are 32 keywords in 'C'.
- It is a portable language.
- It is a structured programming language.
- It can extend itself because of large number of built in function.
- 'C' program can be viewed as a group of building blocks called function.

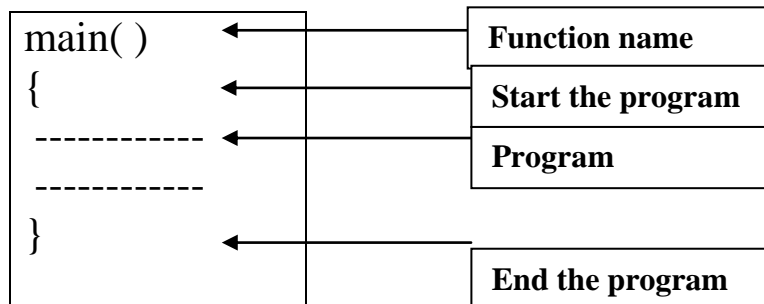
❖ The main function:

```
main( )
```

```
void main( )
```

It has a empty pair of parentheses and it have a no arguments.

void is a keyword.



❖ Example of 'C' Program:

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

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

```
void main( )
```

← No semicolon

```
{
```

```
    Printf("My College name is Vidhyanagari ");
```

← semicolon

```
    getch( );
```

```
}
```

❖ **Basic Structure of 'C' program:**

Documentation section
Link section
Definition section
Global Declaration
Main() Function { Declaration Part Executable Part }
Sub program section Function.....1 Function.....1 ----- Function.....N

❖ **Documentation Section:**

- It consists of a set of comment line giving the name of the program, the author and other details, which the programmer would like to use later.

How to use Comment?

- /* End terminate with */ // Multiline Comment
- // Singal line Comment
- These statement can be put anywhere in the program.

❖ The link section

- It provides instructions to the compiler to link functions from the system library.
- Ex: #include<stdio.h>
- Ex: #include<conio.h>

❖ The definition section:

- This section is not necessary for all 'c' programs. But whenever we need to declare a constant, we should declare then here instead of declaring in the main program.
- Ex. we need to declare a constant pi which having a fixed value of 3.14 so we can declare.
- Ex: #define pi 3.14

❖ Global declaration section:

- In the C language user used some variable which is used in to more than one function.
- Such variable are called global declaration.
- Ex. void sum(int a, int b)

❖ The main function declaration section:

- Every C program must have specific function named main
- The execution of any program starts at the beginning of its main()
- The Main() is also one UDF
- This section contains two parts:
- Declaration Part: It declares all the variable used in executes part.
- Executable Part: There must be at list one executable part.
- The program execution begins at the opening brace ({) and End with the closing brace (}).

❖ Sub program:

- The sub program section contains all the user defined function that called in the main function.



❖ Editor of 'C' program:

- Start the program
- Write your logic of program
- Save the program '.C' extension.
- Compile the program Alt + F9
- Run the program Ctrl + F9

❖ Character set of 'C' program:

- 'C' allows many characters and symbol.
- Letter: It allows upper case A to Z and lower case a to z.
- Digit: It allows all the decimal 0 to 9.
- Special characters: 'C' allows comma (,), dot (.), colon (:) etc.

❖ C Token: In C program the smallest individual units is called**C Token**

- Keyword
- Operator
- Constant
- Identifier
- String
- Special symbol

❖ Keyword:

- All keywords have fixed meaning and these meaning cannot be changed.
- There are 32 keyword in C compiler.
- Ex:- Auto, double, int, struct, break, else, long, switch, case, enum, register, typedef, char, extern, return, union, const, float, short, unsigned, continue, for, signed, void, default, while, goto, sizeof, volatile, do, if, static.

❖ Identifier:

- Identifier is a name of a variable, structure, union, arrays or function.

❖ Rules for naming identifier:

- The name of identifier can contain letters digits and underscore.
- It must start with an alphabet or even in some cases it can start with underscore. But cannot start with digits.
- The name of identifier can contain maximum 31 characters but without space.(white space)
- Space, comma and any other special symbol not allowed in the name of identifier.
- 'C' language is case sensitive. So upper case and lower case is different.
- Ex. GROWMORE and growmore both are different.

❖ Constants:

- constant in c refer to fixed values that do not change during the execution of a 'C' program. 'C' supports several types of constants as shown below.

✓ Constant

- Numeric constants

✚ Integer

✚ Real

- Character constants

✚ Single character

✚ String

- ✓ Integer: User declare a rounded value

Ex. 23,56,8

- ✓ Real: User declare a floating point value

Ex. 23.45, 56.6

- ✓ Character: User declare a only single character



Ex. 'A', 'a'

- ✓ String character: User declare only group of character.

Ex. "Patel Anand"

❖ What is variable?

- A variable is a data name that may used to store a data value.
- A variable may take different value at different time during execution.
- It consists of letters, digits and the underscore (_) character.

❖ Rules for Naming variable:

- They must begin with a letter(character). Some systems permit underscore as the first character.
- It recognized length 31 character. Length should not be normally more than eight characters.
- Uppercase and lowercase are significant.
- Eg. The variable TEST is not the same as test.
- It should not be a keyword.
- White space are not allowed.

❖ Example of valid and invalid variable:

Valid variable name	Invalid variable name
Height	2fo
sum	my name
marks	While
No	auto



❖ **Data types:**

- C language is very rich in data types.
- The varieties of data types are available in ANSI – C.
- C supports three type of data types,
- Primary (Fundamental or Basic) data type.
- Derived data type.
- User-define data type.
- All C compiler support five type of fundamental data types.
- Integer (int), character (char), floating point (float), double floating point (double) (long double) and null (void).
- Integers are whole numbers with the range of values and define by keyword int.
- Floating point numbers are defining in C by the keyword float, double and long double.
- A single character can be defined as a char data type.
- The void type has no values. This is usually used to specify the type of function.
- ✓ The range of data types is given bellow.

Data Type	Size (in Byte)	Range
✓ char	✓ 1 byte	✓ -127 to 127
✓ unsigned char	✓ 1 byte	✓ 0 to 255
✓ int	✓ 2 byte	✓ -32,768 to 32,767
✓ unsigned int	✓ 2 byte	✓ 0 to 65535
✓ float	✓ 4 byte	✓ 3.4E -38 to 3.4E +38
✓ double	✓ 8 byte	✓ 1.7E -308 to 1.7E +308
✓ long double	✓ 10 byte	✓ 3.4E -4932 to 1.1E +4932
✓ void	✓ Null	✓ Null

Example:-

```
#include <stdio.h>
#include <limits.h>
int main()
{
    int a=10;
    char b='A';
    float c=123.45;
    double d=123.678;
    clrscr();
    printf("\n a=: %d",a);
    printf("\n b=: %c",b);
    printf("\n c=: %f",c);
    printf("\n d=: %ld",d);
    getch();
    return 0;
}
```

❖ Declaration of Variables

- A variable is nothing but a name given to a storage area that our programs can manipulate. Each variable in C has a specific type, which determines the size and layout of the variable's memory; the range of values that can be stored within that memory; and the set of operations that can be applied to the variable.
- The name of a variable can be composed of letters, digits, and the underscore character. It must begin with either a letter or an underscore. Upper and lowercase letters are distinct because C is case-sensitive.

❖ Variable Definition in C:

- A variable definition means to tell the compiler where and how much to create the storage for the variable. A variable definition specifies a data type and contains a list of one or more variables of that type as follows:
- **type variable_list;**
- Here, type must be a valid C data type including char, w_char, int, float, double, bool or any user-defined object, etc., and variable_list may consist of one or more identifier names separated by commas. Some valid declarations are shown here:

```
int i, j, k;  
char c, ch;  
float f, salary;  
double d;
```

- The line `int i, j, k;` both declares and defines the variables i, j and k; which instructs the compiler to create variables named i, j and k of type int.
- Variables can be initialized (assigned an initial value) in their declaration. The initializer consists of an equal sign followed by a constant expression as follows:

```
type variable_name = value;
```

Some examples are:

```
extern int d = 3, f = 5; // declaration of d and f.  
int d = 3, f = 5;      // definition and initializing d and f.  
byte z = 22;           // definition and initializes z.  
char x = 'x';          // the variable x has the value 'x'.
```

❖ Symbolic Constant

- What is a symbolic constant ?
- A symbolic constant is an "variable" whose value does not change during the entire lifetime of the program. Some constants in Mathematics:

Name of the constant	Value of the constant
Pi	3.1415926535...
e (Natural log)	2.718281828...

For example

- , a **C program** consists of the following symbolic constant definitions.
#define PI 3.141593
#define TRUE 1
#define FALSE 0
- #define PI 3.141593 defines a symbolic constant PI whose value is 3.141593. When the program is preprocessed, all occurrences of the symbolic constant PI are replaced with the replacement text 3.141593.
- Note that the preprocessor statements begin with a #symbol, and are not end with a semicolon. By convention, preprocessor constants are written in UPPERCASE.