C Language Introduction-

- **What is Programming?**
- **What is Program?**
- **What is C?**
- **Uses Of C**
- **Features of C / Why we should learn c**
- **How is different from C++**
- **History of C**
- **Generation Of Programming Languages**
- **C Program Structure**
    1. Pre-Processor
    2. Header File
    3. Main() function
    4. Return statement
    5. Comments
- **Tokens in C**
    1. Keywords
    2. Identifiers
    3. String Literals
    4. Constant
    5. Operators
    6. Special Character
- **What is an IDE**
- **What is Compiler**

What is Programming?

Computer Programming is a medium for us to communicate with computers. Just like we use 'Hindi' or 'English' to communicate with each other, Programming is a way for us to deliver our instructions to the computer.

What is Program?

In Computing, a program is a set of instructions to perform specific task.

What is C?

1. C is a powerful general-purpose programming language.
2. It is fast, portable and available in all platforms.

3. If you are new to programming, C is a good choice to start your programming journey.
4. C language was developed by Dennis M. Ritchie at bell laboratory in early 1970s
5. It is one of the most popular computer languages today because of its structure, high-level abstraction, machine independent feature etc.
6. C language was developed to write the UNIX operating system, hence it is strongly associated with UNIX, which is one of the most popular network operating system in use today and heart of internet data superhighway

Uses of C-

1. MySQL database is written using C.
2. C has also been used widely while creating ios and android kernels.
3. Ruby and Pearl are mostly written using C.
4. Most part of the apache and NGINX is written using c.
5. Embedded Systems are created using c.

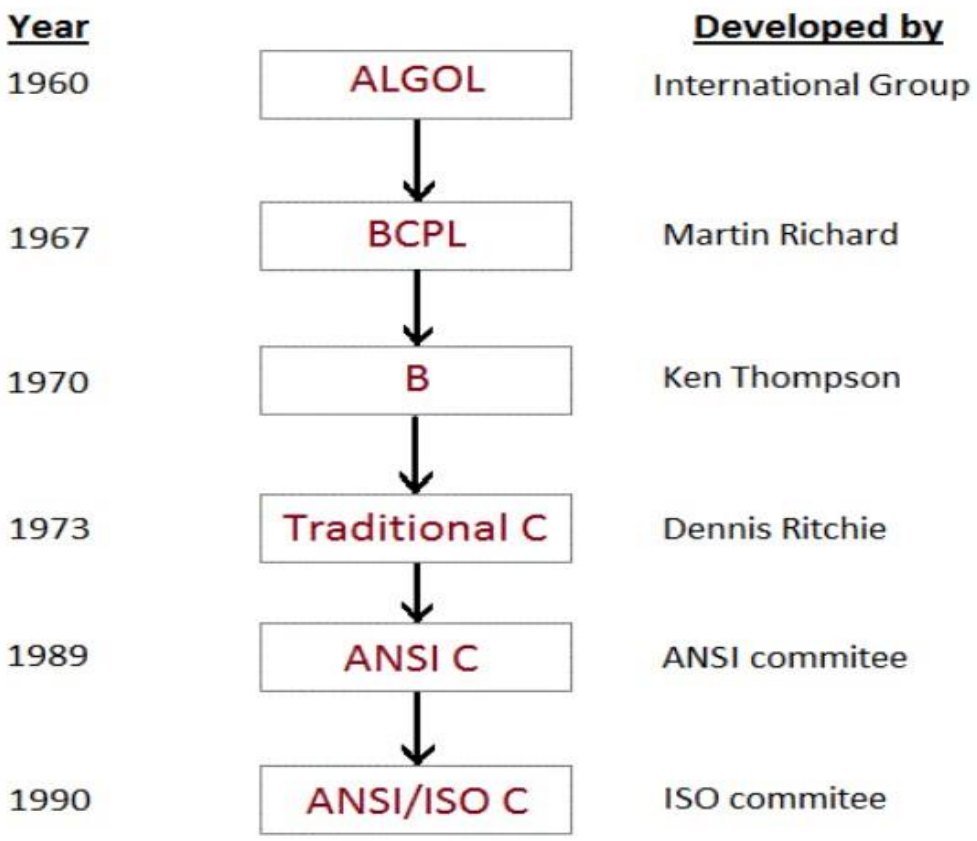
Why should we learn C/ Features of C?

1. Learning any other popular programming language such as Python or C++ becomes a cakewalk already if you know C.
2. C is a flexible language and that gets proven by the fact that it can be used in a variety of applications as well as technologies.
3. C is very fast when compared to other programming languages be it Java or Python.
4. C takes only significant CPU time for interpretation. That is why a lot of Python libraries such as NumPy, pandas, Scikit-learn, etc. are built using C.
5. Being close to Machine language, some of its functions include direct access to machine-level hardware APIs.

How is different from C++?

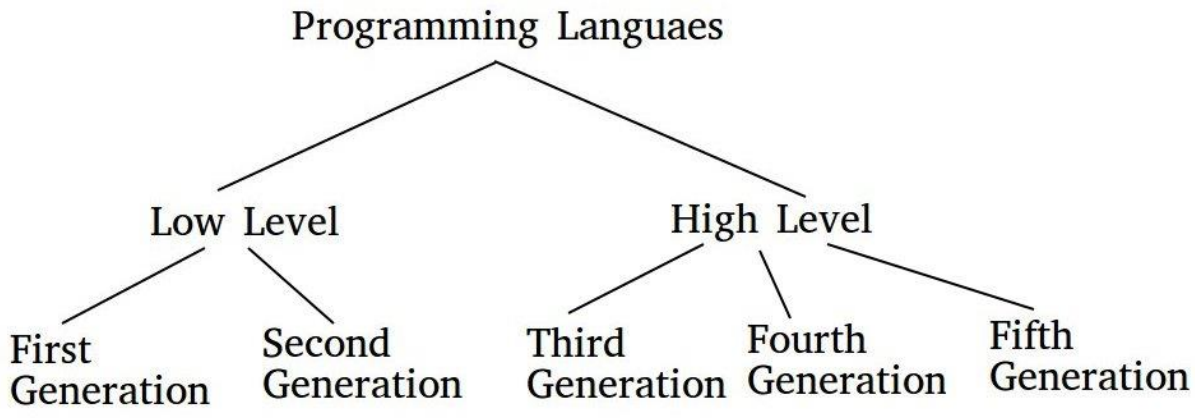
- The syntax of C++ is almost identical to that of C, as C++ was developed as an extension of C.
- In contrast to C, C++ supports classes and objects, while C does not.
- C gives most of the control to the hand of users. Things like memory allocation and manipulation are totally in the hands of the programmer. Being a flexible language, it provides more access to the programmer because of which it is more efficient.
- C is POP(procedure oriented programming) whereas c++ is OOP(Object oriented programming)

History of C Language –



Generation Of Programming Languages :-

There are five generations of Programming languages. They are:



1. First-Generation Language :

The first-generation languages are also called machine languages/ 1G language. This language is machine-dependent. The machine language statements are written in binary code (0/1 form) because the computer can understand only binary language.

2. Second Generation Language :

The second-generation languages are also called assembler languages/ 2G languages. Assembly language contains human-readable notations that can be further converted to machine language using an assembler.

Assembler – converts assembly level instructions to machine-level instructions.

Programmers can write the code using symbolic instruction codes that are meaningful abbreviations of mnemonics. It is also known as low-level language.

3. Third-Generation Language :

The third generation is also called procedural language /3 GL. It consists of the use of a series of English-like words that humans can understand easily, to write instructions. It's also called High-Level Programming Language. For execution, a program in this language needs to be translated into machine language using a Compiler/ Interpreter. Examples of this type of language are C, PASCAL, FORTRAN, COBOL, etc.

4. Fourth Generation Language :

The fourth-generation language is also called a non – procedural language/ 4GL. It enables users to access the database. Examples: SQL, Foxpro, Focus, etc.

These languages are also human-friendly to understand.

5. Fifth Generation Language :

The fifth-generation languages are also called 5GL. It is based on the concept of artificial intelligence. It uses the concept that rather than solving a problem algorithmically, an application can be built to solve it based on some constraints, i.e., we make computers learn to solve any problem. Parallel Processing & superconductors are used for this type of language to make real artificial intelligence.

Examples: PROLOG, LISP, etc.

First C Welcome Program and its Structure:

welcome.c

```
#include<stdio.h>
#include<conio.h>
void main()
{
 clrscr();    //It is use for clear screen
 printf("\n Welcome to Ssi Computer Education");  \\ \n(New line Character)  use for new line
 printf("\n C & C++ Programming Language  ");
 getch();    // It is use for get character from Keyboard
}
```

Basic Structure & Syntax

All C Programs have to follow a basic structure.

A C Program Starts with a main function and executes instructions present inside it.

Each instructions is terminated with a semicolon (;).

There are some rules which are applicable to all the c programs:

1. Every program execution starts from main() function.
2. All the statements are terminated with a semicolon.
3. Instructions are case sensitive.
4. Instructions are executed in the same order in which they are written.

Programming in C involves following a basic structure throughout. Here's what it can be broken down to.

- Pre-Processor Commands
- Header files
- Function
- Variables
- Statements & Expressions
- Comments

All these are essential parts of a C language Program.

Pre-processor commands

Pre-processor commands are commands which tell our program that before its execution, it must include the file name mentioned in it because we are using some of the commands or codes from this file.

They add functionalities to a program.

One example could be,

#include <math.h>

We include math.h to be able to use some special functions like power and absolute. #include<filename.h> is how we include them into our programs.

Header files:

- Collection of predefined/built in functions developed.
- It is always declares on heading side of program hence it is called header file
- It is identified with the extension(.h)
- It gets installed while installing IDE(integrated development environment)
- It stores functions as per their categories hence they are called library

Syntax

An example below shows how a basic C program is written.

```
Declaration of header file          //name of the header files of which functions are been used
main()                             /*it is called main function which stores the execution of program*/
{                                                 //start of the program
            //program statements
}                                                //end of the program
```

- Here, the first line is a pre-processor command including a header file stdio.h.
- C ignores empty lines and spaces.
- There is a main() function then, which should always be there.

main() function

- main() function is a function that must be there in every C program.
- Everything inside this function in a C program will be executed.
- In the above example, void written before the main() function is the return type of main() function.
- The curly braces { } just after the main() function encloses the body of main() function.

Comments

- We can add comments in our program to describe what we are doing in the program.
- These comments are ignored by the compiler and are not executed.
- To add ***a single line comment***, start it by adding two forwarded slashes // followed by the comment.
- To add ***multi line comment***, enclose it between /*…*/, just like in the program above.

Return Statement – return 0;

- A return statement is just meant to define the end of any C Program.
- All the C program can be writted and edited in normal text editors like Notedpad or Notepad++ and must be saved with a file name with extension as .c if you do not add the extension .c then the compiler will not recognize it as a C language Program file.

Library Functions:

C language has a lot of valuable library functions which is used to carry out certain tasks. For instance printf function is used to print values on the screen.

Printf("This is %d",i);

%d for integers

%f- for real values

%c for characters

Tokens in C

- Tokens in C is the most important element to be used in creating a program in C.
- We can define the token as the smallest individual element in C.

- For `example, we cannot create a sentence without using words; similarly, we cannot create a program in C without using tokens in C.
- Therefore, we can say that tokens in C is the building block or the basic component for creating a program in C language.

A C Program is made up of different tokens combined, these tokens include:

- Keywords
- Identifiers
- Constants
- String Literal
- Operators
- Special Characters

Keywords: -

- Keywords are reserved words that can not be used elsewhere in the program for naming a variable or a function.
- They have a specific function or task and they are solely used for that. Their functionalities are pre-defined.
- One such example of a keyword could be return which is used to build return statements for functions. Other examples are auto, if, default, etc.
- Whenever we write any keyword in IDE their colour slightly changes and it looks different from other variables or functions for example in turbo c all keywords are turns into white colour.

Identifiers:-

- Identifiers are names given to variables or functions to differentiate them from one another.
- Their definitions are solely based on our choice but there are a few rules that we have to follow while naming identifiers. One such rule says that the name can not contain special symbols such as @, -, *, <, etc.
- C is a case-sensitive language so an identifier containing a capital letter and another one containing a small letter in the same place will be different. For example, the three words: Code, code, and cOde can be used as three different identifiers.
- Rules for naming identifier-
    1. One should not name any identifier starting with numeric value or symbol. It should start only with underscore or alphabet
    2. They should not contain space
    3. Giving logical names is recommended as per our program

Constants:-

- Constants are very similar to a variable and they can also be of any data type.
- The only difference between a constant and a variable is that a constant's value never changes

String Literal :-

- String literals or string constants are a sequence of characters enclosed in double quotation marks.
- For example, "This is a string literal!" is a string literal. C method printf() utilizes the same to format the output.

Operators:

- Operators in C is a special symbol used to perform the functions.
- he data items on which the operators are applied are known as operands.
- Operators are applied between the operands

Special Characters in C

Some special characters are used in C, and they have a special meaning which cannot be used for another purpose.

- *Square brackets [ ]:* The opening and closing brackets represent the single and multidimensional subscripts.
- *Simple brackets ( ):* It is used in function declaration and function calling. For example, printf() is a pre-defined function.
- *Curly braces { }:* It is used in the opening and closing of the code. It is used in the opening and closing of the loops.
- *Comma (,):* It is used for separating for more than one statement and for example, separating function parameters in a function call, separating the variable when printing the value of more than one variable using a single printf statement.
- *Hash/pre-processor (#):* It is used for pre-processor directive. It basically denotes that we are using the header file.
- *Asterisk (*):* This symbol is used to represent pointers and also used as an operator for multiplication.
- *Tilde (~):* It is used as a destructor to free memory.
- *Period (.):* It is used to access a member of a structure or a union.

Requirement before you start –

To start using c, you need two things:

- A text editor, like Notepad, or an IDE, like VSCode to act as a platform for you to write C code
- A compiler, like GCC to translate the C code you have written which is a high-level language into a low-level language that the computer will understand.

What is an IDE?

- IDE stands for an Integrated Development Environment.
- It is nothing more than an enhanced version of a text editor that helps you write more efficient and nicer code.
- It helps to differentiate different parts of your codes with different colors and notifies you
- if you are missing some semicolon or bracket at some place by highlighting that area.
- A lot of IDEs are available, such as DEVC++ or Code Blocks, but we will prefer using VS Code for this tutorial series.

What is the Compiler?

- A compiler is used to run the program of a certain language which is generally high-level by converting the code into a language that is low-level that our computer could understand.