

Lecture #1:

Introduction To C

Definition:

C is a procedural programming language. It was developed by **Dennis Ritchie** in the year 1972. at Bell Laboratories (USA).



It is a General-purpose, **Structured**, **Machine Independent**, Simple and Flexible programming language. It was mainly developed as a system programming language to write an operating system.

Features of C Programming Language

- **High-Level Language**

C provides strong **abstraction** in case of its libraries and built-in features that make it machine-independent. It is capable enough to develop system applications such as the kernel, driver, etc.

- **Structured Language**

C Language supports structured programming which includes the use of **functions**. Functions reduce the code complexity and are totally reusable.

- **Rich Library**

Unlike its predecessors, C language incorporates multiple built-in **arithmetic** and **logical** functions along with many built-in libraries which make development faster and convenient.

- **Extensible**

C Language is a High-level language and is also open for **upgrades**. Hence, the programming language is considered to be **extensible** like any other high-level languages.



- **Recursion**

C Language supports function **back-tracking** which involves recursion. In the process of recursion, a function is called within another function for multiple numbers of times.

- **Pointers**

C enables users to directly interact with memory using the **pointers**. We use pointers in memory, structure, functions, arrays, stack and many more.

- **Faster**

C Language comes with a minimal number of **libraries** and **built-in** functions which makes the compile and execution-times to be less and the system faces low overhead.

- **Memory Management**

C provides the best in class memory management. It can both allocate and deallocate memory dynamically. the **malloc ()**, **calloc ()**, **realloc ()** functions are used to allocate memory dynamically and **free ()** function is used to deallocate the used memory at any instance of time.

Structure:

By structure, it is meant that any program can be written in this structure only. Writing a C program in any other structure will hence lead to a Compilation Error.

```
#include<stdio.h>

int main()

{

int a=10;

printf("%d",a);

return 0;

}
```

A. Header File Inclusion:

The first and foremost component is the inclusion of the Header files in a C program. A header file is a file with extension .h which contains C function declarations and macro definitions to be shared between several source files.

Syntax to include a header file in C:

```
#include
```

B. Main Method Declaration:

The next part of a C program is to declare the main() function. The syntax to declare the main function is:

Syntax to Declare main method:

```
int main(){ }
```

C. Variable Declaration:

The next part of any C program is the variable declaration. It refers to the variables that are to be used in the function. Please note that in the C program, no variable can be used without being declared. Also in a C program, the variables are to be declared before any operation in the function.

Example:

```
int main()
{
int a;
.
.
}
```

D. Body:

Body of a function in C program, refers to the operations that are performed in the functions. It can be anything like manipulations, searching, sorting, printing, etc.

Example:

```
int main()
{
    int a;
    printf("%d", a);
    .
    .
```

E. Return Statement:

The last part in any C program is the return statement. The return statement refers to the returning of the values from a function. This return statement and return value depend upon the return type of the function. For example, if the return type is void, then there will be no return statement. In any other case, there will be a return statement and the return value will be of the type of the specified return type.

Example:

```
int main()
{
    int a;
    printf("%d", a);
    return 0;
}
```

Source: [GeeksForGeeks](#) || [CodeCogs Contributors](#).

CodeCogs