Introduction to C Language

History:

In 1960s a number of computer languages were used for various purpose. e.g. COBOL [Common Business Oriented Language] for commercial applications. FORTRAN for engineering and scientific application.

There was a need of single language which could be used to develop all type of applications.

1. First developed Algol60 in 1960. But it was too abstract and too general.

2. Then CPL [Combined Programming Language] was developed in 1963 at Cambridge University but it was too difficult to learn.

3. Then Martin Richards developed BCPL [Basic Combined Programming Language] in 1967 at Cambridge University but it was less powerful and too specific.

4. Then Ken Thompson developed B in 1970 at AT & T's Bell Labs, for first Unix system. But it was very specific.

5. Then Dennis Ritchie designed C in 1972 by taking B and BCPL as base. C consists of number of data types. Initially C was developed for and implemented on UNIX OS.

Features Of ‘C’ Programming Language:

C language is one of the powerful languages. Below are some of the features of C language.

- Reliability
- Portability
- Flexibility
- Interactivity
- Modularity
- Efficiency and Effectiveness

Advantages:

1. Powerful and efficient language
   C is a robust language as it contains many data types and operators to give you a vast platform to perform all kinds of operations.

2. Portable language
   C is very flexible, or we can say machine independent that helps you to run your code on any machine without making any change or just a few changes in the code.

3. Built-in functions
   There are only 32 keywords in ANSI C, having many built-in functions. These functions are helpful when building a program in C.
4. **Structured programming language**
   C is structure-based. It means that the issues or complex problems are divided into smaller blocks or functions. This modular structure helps in easier and simpler testing and maintenance.

5. **Middle-level language**
   C is a middle-level programming language that means it supports high-level programming as well as low-level programming. It supports the use of kernels and drivers in low-level programming and also supports system software applications in the high-level programming language.

6. **Procedural programming language**
   C follows a proper procedure for its functions and subroutines. As it uses procedural programming, it becomes easier for C to identify code structure and to solve any problem in a specific series of code. In procedural programming C variables and functions are declared before use.

7. **Dynamic memory allocation**
   C provides dynamic memory allocation that means you are free to allocate memory at run time. For example, if you don't know how much memory is required by objects in your program, you can still run a program in C and assign the memory at the same time.

**Limitations:**
1. C does not have concepts of object oriented programming, that's why C++ is developed.
2. There is no runtime checking in C language.
3. Lack of exception handling

**Structure of C Program:**

```
Documentation Section
Link Section
Definition Section
Global Declaration Section
Function Section
main()
{
    Variable declaration-initialization;
    Program statements;
}
Subprogram Section
Function1()
...
Function2()
```

1. **Documentation Section** (optional): Consist of comments. Which are used to convey program information and other details. Comments are not executed by compiler.

   Single line comments - //
Multiline comments - /*.......*/

2. **Link Section**: Consist of pre-processor directives which link the compiler to the standard library functions in our program for that we need to include appropriate header files.

3. **Definition Section** (optional): Defines all symbolic constants like `#define PI 3.14`

4. **Global Declaration Section** (optional): Declares global variables. These variables can be used in more than one function and so they are declared outside of all functions.

5. Every C program must have one main() function. C program execution start with main().

7. C program contains zero or more variable declaration and statements.

8. **Subprogram Section** (optional): Contains all user defined functions that are called in main().

<table>
<thead>
<tr>
<th>#include&lt;x.h&gt;</th>
<th>#include&quot;x.h&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>It searches file only in standard C library location</td>
<td>It first searches in current dictionary and if not found then searches in standard C library location.</td>
</tr>
<tr>
<td>This is used to include standard library files.</td>
<td>This is used to include user defined header files.</td>
</tr>
</tbody>
</table>

Example:

1) ```c
#include<stdio.h>
#define PI 3.14
float r=8.3;
main()
{
    float ans=0;
    ans=PI*r*r;
}
```  

2) ```c
#include<stdio.h>
#define MAX(x,y)  ((x>y)?x:y)
main()
{
    int a=20,b=5,ans;
    ans=MAX(a,b);
    printf("\nMaximum : %d",ans);
} ```