UNIT 3
BASIC CONCEPT OF C

LH – 5HRS

PRESENTED BY:

ER. SHARAT MAHARJAN

C PROGRAMMING

CONTENTS (LH -5HRS)

3.1 Introduction, History, Features, Advantages and Disadvantages, Structure of C Program,

3.2 Compiling Process, C Preprocessor and Header Files,

3.3 Library Function, Character Set, Comments,

3.4 Tokens and its types,

3.5 Data types,

3.6 Escape Sequences, Preprocessor Directive

3.1 Introduction, History, Features, Advantages and Disadvantages, Structure of C Program

1. Introduction

- C is a procedural programming language. It was initially developed by Dennis Ritchie between 1969 and 1973 at Bell Labs.
- It is one of the most popular procedural, general purpose, and high-level programming language used in all operating systems of today.
- The main features of C language include low-level access to memory, simple set of keywords, and clean style. These features make C language suitable for system programming like operating system or compiler development.
- Although C was designed for writing system software, it is also widely used for developing application software.
- C has greatly influenced many other popular programming languages, most notably C++, which began as an extension to C.
- C supports the programmer with a rich set of built-in functions and operators. C supports different types of statements like sequential, selection, looping etc.
- Procedural programming concept is well supported in C which helps in dividing the programs into functional modules or code blocks.

2. History of C

In 1822 with the invention of differential engine by Charles Babbage, the necessity of programming language arises. In 1949, Von Neumann developed short code language i.e. mnemonics. Till 1960s there were numbers of programming language but all were special purposed like FORTRAN (Formula Translation) and COBOL (Common Business-Oriented Language).

After special purpose language, the necessity of general purpose language arised and development of ALGOL (Algorithm language) and then CPL (Combined Programming Language) occurred. CPL was jointly developed by university of Cambridge and university of London, the next language based on CPL is called BCPL (Basic CPL) designed by Martin Richards of University of Cambridge in 1966, which was the intended as system programming language, used to write compilers. Around the same time, a language called B was developed at Bell Labs by Ken Thompson with contribution of Dennis Ritchie in 1969. The name B was originated from Bonnie (Thompson's wife). Since, B also turned to be very specific, in 1972; C was written and developed by Dennis Ritchie at Bell Labs. It was named C because most of the features are derived from B language. C program is general purpose, structured, procedural programming language. In 1980, ANSI (American National Standard Institute) standardized C language.

3. Features of C Language

a. **Simple:** C language is a simple and easy programming language for both learning and programming. Most of the computer programming languages follow its syntax and concepts, like C++, Java, C# etc.

b. **Structured:** C language is a structured programming language. In this language, we can divide a big program into a several different structured blocks and manage the program easily.

c. **Portable:** C language is a portable programming language due to which, we can run C language programs on different platforms with little modification or without modification.

d. **Middle Level:** C language is a middle-level programming language, because it contains features of both low-level and high-level programming languages.

e. **Fast execution:** The compilation and execution of C programs is much faster than any other language.

f. **Rich Set Library:** C language provides lot of libraries like stdio.h, math.h, string.h, conio.h, graphics.h etc. These libraries help us to create programs or applications easily and faster.

g. **Memory management:** C language provides better memory management using pointer. We can allocate dynamic memory at runtime in C using pointer functions like malloc(), calloc(), alloc() etc.

h. **Modular:** C language is a modular programming language which means we can divide a C program into small modules according to functionality; that makes the program easy to understand.

4. Advantages and Disadvantages of C Language:

Advantages:

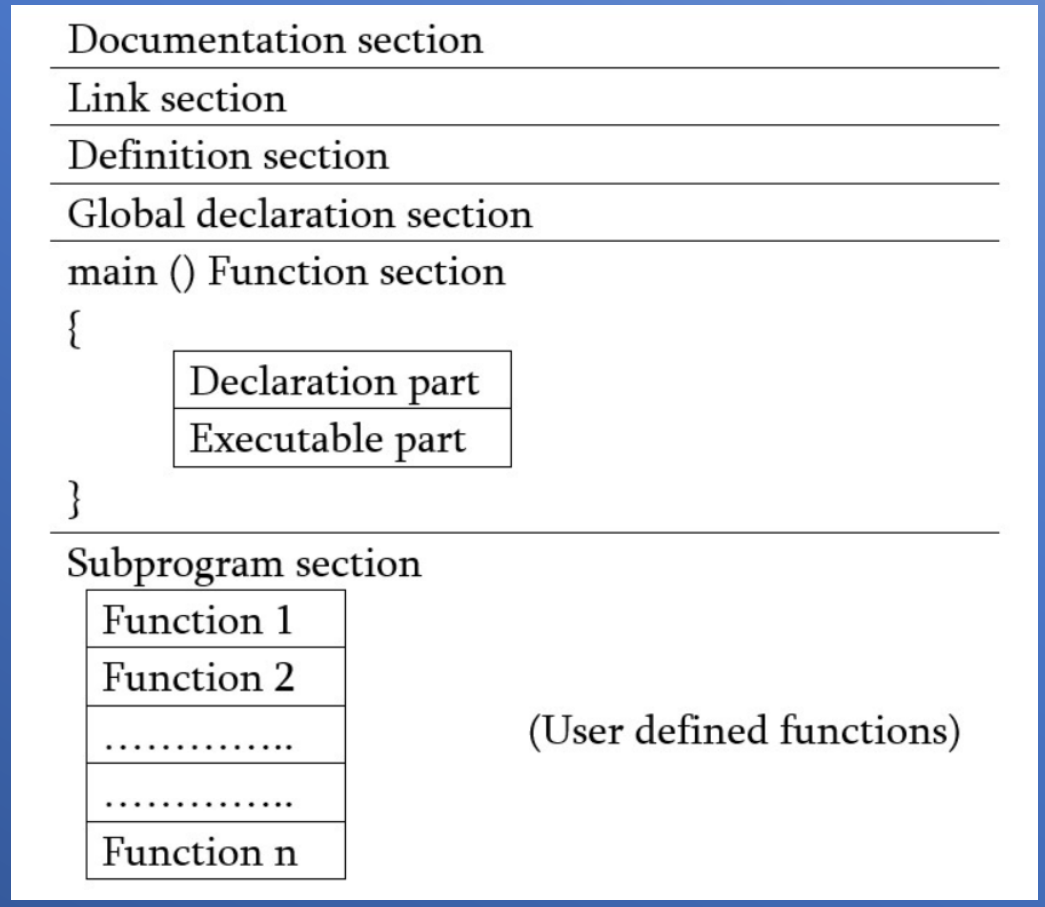
- C is very easy to learn language for expressing ideas in programming in a comfortable way.
- C is close to machine because it supports features like pointers, bytes and bit level manipulation.
- C is structured programming language which makes the program easier to understand and modify.
- C language produces portable programs, they can be run on any compiler with little or no modifications.
- C is powerful programming language as it provides a wide variety of inbuilt data types and ability to create custom data types using structures.

Disadvantages:

- C doesn’t have concept of OOP’s, that’s why C++ is developed.
- There is no runtime checking in C language.
- There is no strict type checking. For example, we can pass an integer value for the floating data type.
- C doesn’t have the concept of namespace.
- C doesn’t have the concept of constructor or destructor.

5. Structure of C Program

A C-program has the following basic structure:



Documentation section:

- It consists of a set of comment lines giving the name of the program, the author and other details as a short description of the purpose of the program etc.

Link section:

- This section provides instructions to the compiler to link functions from the system library.

Definition section:

- This section defines all the symbolic constants.

Global Declaration section:

- The variables which are used in more than one functions or blocks are called global variables. These variables are defined or declared in this section. Declaring the variables before the main function makes the variables accessible to all the functions in a C language program. Declaring the variables within main functions or other functions makes the usage of the variables confined to the function only and not accessible outside. In this section, all the user defined functions are also declared.

main () Function section:

- Every C program must have one main function. It contains different declaration and executable statements between the opening and closing braces.

Subprogram section:

- It contains all the user-defined functions that are called in other and main function. User-defined functions are generally places immediately after the main function although they may appear in any order.

Example:

```
/*Program to find area and circumference of a circle*/        →Documentation section
#include<stdio.h>          → Link section
#define PI 3.14            → Definition section
float r,a,c;
float area();                          Global Declaration section
float circumference();
int main(){
        printf("Radius= ");
        scanf("%f",&r);
        a=area();
        c=circumference();                          main () function section
        printf("Area= %f",a);
        printf("\nCircumference= %f",c);
        return 0;
}
float area(){
        return PI*r*r;                  sub-function1 ()
}
float circumference(){
        return 2*PI*r;                  sub-function2 ()
}
```

OUTPUT:

```
Radius= 7
Area= 153.860001
Circumference= 43.959999
```

3.2 Compiling Process, C Preprocessor and Header Files

1. Compiling Process:

- The code of the program written in high level language is called source code; which is not directly understandable by the computer. Hence, the source code should be translated into computer understandable form i.e. machine level language.
- The translation process is called compilation; which is done by special software called compiler. The machine level code that we set after compilation is called object code which is the executable file that is compatible with particular platform (windows, linux etc.)
- Execution of the program involves loading the executable object code into the computer memory and executes the instructions. While executing the program, the program may load data from memory or keyboard.

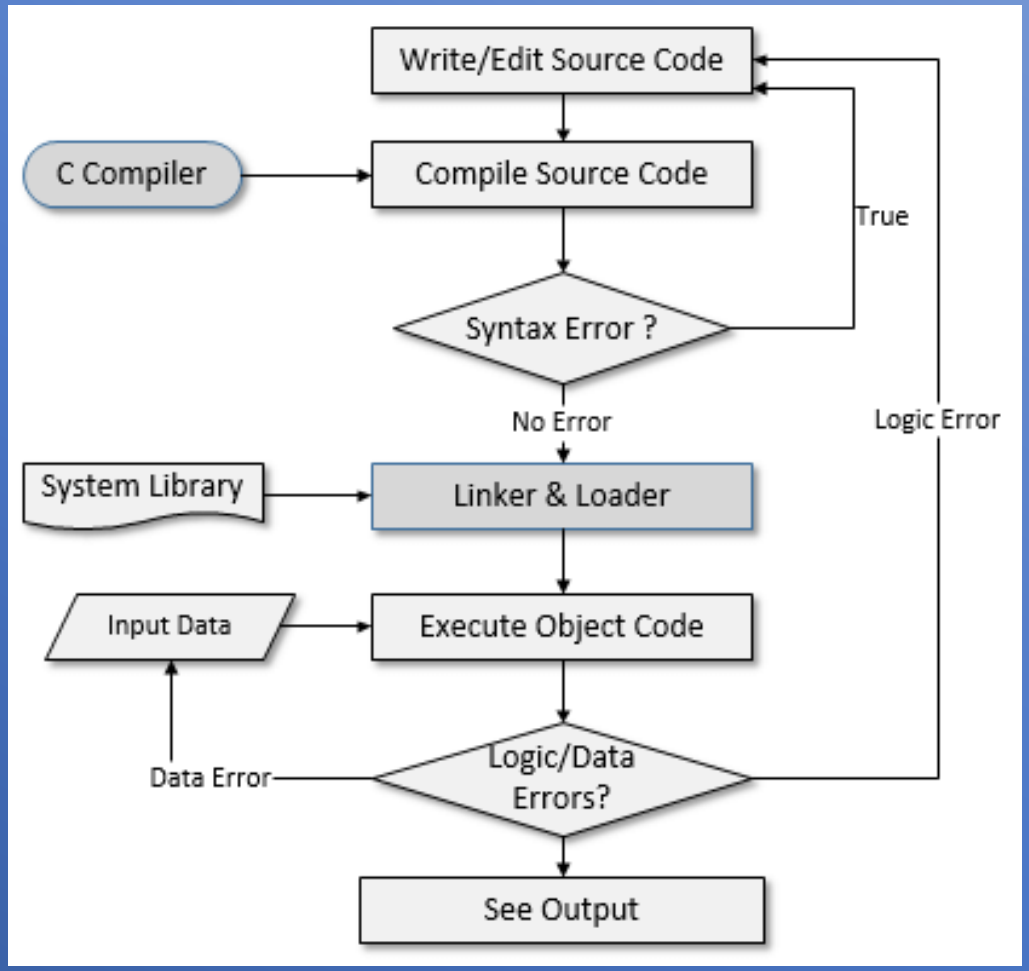


Figure: Process of compiling and running a C program

2. C preprocessor and Header Files:

- The C preprocessor is a **macro processor** that is used automatically by the C compiler to transform our program before actual compilation. It is called a macro processor because it allows us to define **macros**, which are brief abbreviations for longer constructs.
- The C preprocessor provides four separate facilities that we can use as:
  - Inclusion of header files. These are files of declarations that can be substituted into our program.
  - Macro expansion. We can define **macros**, which are abbreviations for arbitrary fragments of C code, and then the C preprocessor will replace the macros with their definitions throughout the program.
  - Conditional compilation. Using special preprocessing directives, we can include or exclude parts of the program according to various conditions.
  - Line control. If we use a program to combine or rearrange source files into an intermediate file which is then compiled, we can use line control to inform the compiler of where each source line originally came from.

- The file with .h extension which defines several built-in functions like printf(), scanf(), clrscr(), getch() is called **header file**. Simply, it is the collection of predefined functions.
- For example: stdio.h, conio.h, math.h etc.
- The inclusion of header files in the program is the pre-information to the compiler regarding the library functions used and its relevant codes to be executed.
- For example: printf() function is defined in stdio.h header file. To use this function in program, the corresponding header file in which it is define (i.e. stdio.h) should be included in the program as #include<stdio.h>.

3.3 Library Function, Character Set, Comments

1. Library Function:

- Library functions are built-in functions that are grouped together and placed in a common location called library.
- Each function here performs a specific operation. We can use this library functions to get the pre-defined output.
- All C standard library functions are declared by using many header files. These library functions are created at the time of designing the compilers.
- We include the header files in our C program by using **#include<filename.h>**. Whenever the program is run and executed, the related files are included in the C program.

Header File Functions

- Some of the header file functions are as follows −
  - **stdio.h** − It is a standard i/o header file in which Input/output functions are declared
  - **conio.h** − This is a console input/output header file.
  - **string.h** − All string related functions are in this header file.
  - **stdlib.h** − This file contains common functions which are used in the C programs.
  - **math.h** − All functions related to mathematics are in this header file.
  - **time.h** − This file contains time and clock related functions. Built functions in stdio.h

2. Character Set:

- The set of characters that are used to form words, numbers and expressions in C is called C character set. The combination of these characters form words, numbers and expressions.

The C character set is grouped into the following four categories:

a. Letters or alphabets
b. Digits
c. Special Characters
d. White Spaces

a. Letters

- Uppercase: A….Z
- Lowercase: a….z

b. Digits

- All decimal digits: 0….9

c. Special Characters

- ~ @ # $ % ^ & * ( ) _ - + = { } [ ] ; : ' " / ? . > , < \ |

d. White Spaces

- Blank space
- Horizontal tab
- New line
- Carriage return
- Form feed

3. Comments:

- The statements written in program which are not compiled and executed are called comments. They are used for documentation purpose. They are ignored by compiler and user is free to write his/her comment within those symbols. There are two types of comments, they are

a. Single Line Comment

It means we can give comment for a single line. For example:

```
//C programming is more interesting.
```

Here // is used as a single line comment, it has no ending symbol.

b. Multiple Line Comment

It means user can give comment for multiple lines with between the symbols /* */. This is applicable in both C and C++. For example:

```
/* C programming
        has rich
        set of operators*/
```

3.4 Tokens and its types

- Every C program is a collection of instructions and every instruction is a collection of some individual units. Every smallest individual unit of a c program is called token. Every instruction in a c program is a collection of tokens. Tokens are used to construct c programs and they are said to the basic building blocks of a C program.

In a C program tokens may contain the following...

1. Identifiers
2. Keywords
3. Operators
4. Special Symbols
5. Constants
6. Strings

Identifiers:

- An identifier is a collection of characters which acts as the name of variable, function, array, pointer, structure, etc.
- The identifier is a user-defined name of an entity to identify it uniquely during the program execution.

Example

```
int marks;
char studentName[30];
```

- Here, marks and studentName are identifiers.

Rules for Creating Identifiers:

1. An identifier can contain **letters** (UPPERCASE and lowercase), **numerics** & **underscore** symbol only.
2. An identifier should not start with a numerical value. It can start with a letter or an underscore.
3. We should not use any special symbols in between the identifier even whitespace. However, the only underscore symbol is allowed.
4. Keywords should not be used as identifiers.
5. There is no limit for the length of an identifier. However, the compiler considers the first 31 characters only.
6. An identifier must be unique in its scope.

Keywords:

- Keywords are the reserved words with predefined meaning which already known to the compiler.
- In the C programming language, there are 32 keywords. All the 32 keywords have their meaning which is already known to the compiler.
- Whenever C compiler come across a keyword, automatically it understands its meaning.

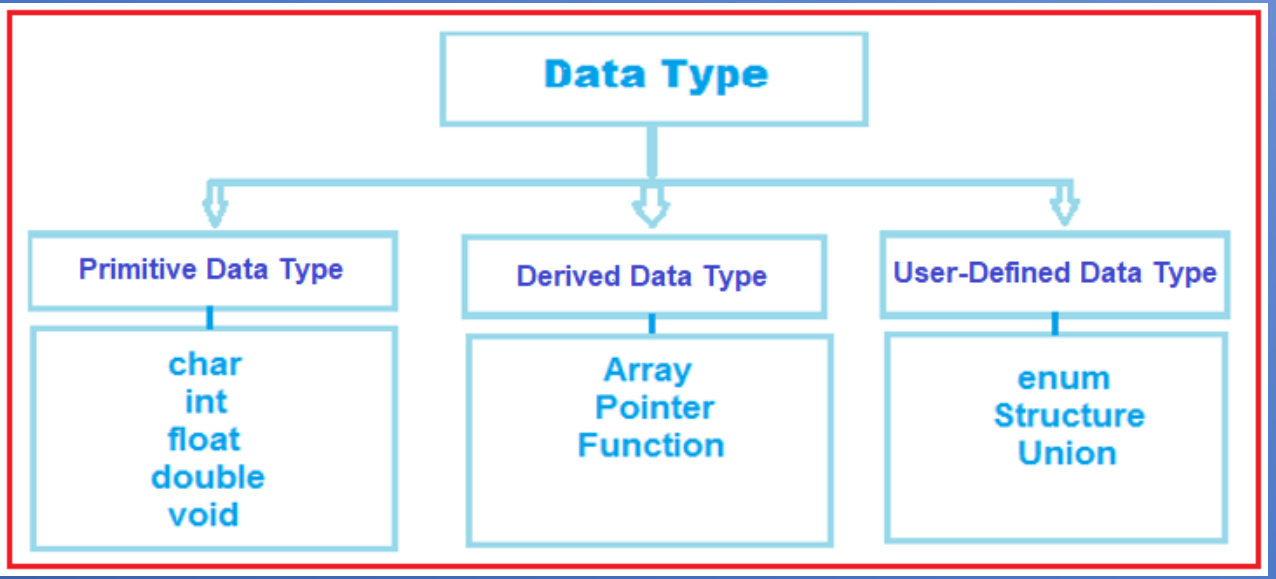
auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

3.5 Data types

- The Data type is a set of value with predefined characteristics.
- Data types are used to declare variable, constants, arrays, pointers and functions.

C language data types can be classified as:

1. Primary/Primitive data type
2. Derived data type
3. User-Defined data type



Data type	Size(bytes)	Range	Format String
char	1	-128 to 127	%c
unsigned char	1	0 to 255	%c
short	2	-32,768 to 32,767	%d
unsigned short	2	0 to 65535	%u
int	2	–32,768 to 32,767	%d
unsigned int	2	0 to 65535	%u
long	4	-2147483648 to +2147483647	%ld
Unsinged long	4	0 to 4294967295	%lu
float	4	3.4e-38 to +3.4e-38	%f
double	8	1.7 e-308 to 1.7 e+308	% lf
long double	10	3.4 e-4932 to 1.1 e+4932	%lf

3.6 Escape Sequences, Preprocessor Directive

Escape Sequences:

- In C, all escape sequences consist of two or more characters, the first of which is the backslash, \ (called the "Escape character"); the remaining characters determine the interpretation of the escape sequence.
- For example, \n is an escape sequence that denotes a newline character.

Constant	Meaning
\a	Alert
\b	Backspace
\f	Form feed
\n	New line
\r	Carriage Return
\t	Horizontal tab
\v	Vertical tab
\'	Single quote
\"	Double quote
\?	Question mark
\\	Backslash
\0	Null

Pre-processor Directives:

- These are placed in the source program before the main function. When source code is compiled, it is examined by the pre-processor for any pre-processor directives. If there are any, appropriate actions are taken and then the source program is handed over to the compiler.
- They follow special syntax rules: they all begin with the symbol # (hash) and do not require a ; (semicolon) at the end.

For example:

#include<stdio.h> #define PI 3.14

THANK YOU FOR YOUR ATTENTION