UNIT 1 BASICS OF C PROGRAMMING **6**

Problem Solving Techniques: Introduction to Algorithm, Pseudo code, Flow Chart, Structure of 'C' program. C Tokens: Keywords, Data Types, Constants, Variables - Declaration - Qualifiers – typedef

1.2 PROBLEM C PROGRAMMING

C is a general-purpose programming language created by Dennis Ritchie at the Bell Laboratories in 1972. It is a very popular language. It is a fundamental language in the field of computer science. C language combines the power of a **low-level language** and a **high-level language**. The low-level languages are used for system programming, while the high-level languages are used for application programming. It is because such languages are flexible and easy to use. Hence, C language is a widely used computer language.

It supports various operators, constructors, data structures, and loop constructs. The features of C programming make it possible to use the language for system programming, development of interpreters, compilers, operating systems, graphics, general utilities, etc. C is also used to write other applications, such as **databases, compilers, word processors,** and **spreadsheets**.

1.1 STRUCTURE OF C PROGRAM

The structure of a language gives us a basic idea of the order of the sections in a program. We get to know when and where to use a particular **statement, variable, function, curly braces, parentheses**, etc. It also increases our interest in that programming language.

Thus, the structure helps us analyse the format to write a program for the least errors. It gives better clarity and the concept of a program.

Sections of a C program

The sections of a C program are listed below:

1. Documentation section
2. Preprocessor section
3. Definition section

4. Global declaration
5. Main function
6. User defined functions

1. Documentation section

It includes the statement specified at the beginning of a program, such as a program's **name, date, description,** and **title**. Comment lines are used. It is represented as:

//name of a program

OR

/* Overview of the code */

Both methods work as the document section in a program. It provides an overview of the program. Anything written as comments will be considered a part of the documentation section and will not interfere with the specified code.

2. Preprocessor section

The preprocessor section contains all the header files used in a program. It informs the system to link the header files to the system libraries. It is given by:

#include<stdio.h>
#include<conio.h>

The **#include** statement includes the specific file as a part of a function at the time of the compilation. Thus, the contents of the included file are compiled along with the function being compiled.

The **#include<stdio.h>** consists of the contents of the standard input output files, which contains the definition of stdin, stdout, and stderr. Whenever the definitions stdin, stdout, and stderr are used in a function, the statement #include<stdio.h> need to be used.

There are various header files available for different purposes. For example, **# include <math.h>.** It is used for mathematic functions in a program.

3. Define section

The define section comprises of different constants declared using the define keyword. It is given

by:

```
#define a = 2
```

4. **Global declaration**

The global section comprises of all the global declarations in the program. It is given by:

```
float num = 2.54;
int a = 5;
char ch ='z';
```

5. Main function

main() is the first function to be executed by the computer. It is necessary for a code to include the main(). It is like any other function available in the C library. Parenthesis () are used for passing parameters (if any) to a function.

The main function is declared as: main()

We can also use int or void with the main (). The void main() specifies that the program will not return any value. The int main() specifies that the program can return integer type data.

```
int main()
OR
void main()
```

Main function is further categorized into **local declarations, statements,** and **expressions.**

6. Local declarations

The variable that is declared inside a given function or block refers to as local declarations.

```
main()
    {
int i = 2;
i++;
}
```

7. Statements

The statements refer to **if, else, while, do, for**, etc. used in a program within the main function.

Expressions: An expression is a type of formula where operands are linked with each other by the use of operators. It is given by:

a - b;

a +b;

8. User defined functions

The user defined functions specified the functions specified as per the requirements of the user. For example, color(), sum(), division(), etc.

The program (basic or advance) follows the same sections as listed above.

Return statement is generally the last section of a code. But, it is not necessary to include. It is used when we want to return a value. The return function returns a value when the return type other than the void is specified with the function.

Return type ends the execution of the function. It further returns control to the specified calling function. It is given by

return;

Or

return expression;

For example,

return 0;

Example 1: To find the sum of two numbers given by the user

```
/* Sum of two numbers */
#include<stdio.h>
int main()
{
```

```
int a, b, sum;
printf("Enter two numbers to be added ");
scanf("%d %d", &a, &b);
// calculating sum
sum = a + b;
printf("%d + %d = %d", a, b, sum);
return 0;  // return the integer value in the sum
}
```

Output

```
Enter two numbers to be added 3 5
3 + 5 = 8
```

The detailed explanation of each part of a code is as follows:

/* Sum of the two numbers */	It is the comment section. Any statement described in it is not considered as a code. It is a part of the description section in a code. The comment line is optional. It can be in a separate line or part of an executable line.
#include<stdio.h>	It is the standard input-output header file. It is a command of the preprocessor section.
int main()	main() is the first function to be executed in every program. We have used int with the main() in order to return an integer value.
{… }	The curly braces mark the beginning and end of a function. It is mandatory in all the functions.
printf()	The printf() prints text on the screen. It is a function for displaying constant or variables data. Here, 'Enter two numbers to be added' is the parameter passed to it.

scanf()	It reads data from the standard input stream and writes the result into the specified arguments.
sum = a + b	The addition of the specified two numbers will be passed to the sum parameter in the output.
return 0	A program can also run without a return 0 function. It simply states that a program is free from error and can be successfully exited.

Example 2: Below C program to find the sum of 2 numbers using function

```
        // Documentation
#include <stdio.h>
        // Definition
#define X 20
        // Global Declaration
int sum(int y);
        // Main() Function
int main(void)
{
  int y = 55;
  printf("Sum: %d", sum(y));
  return 0;
}
        / Subprogram
int sum(int y)
{
  return y + X;
}
```

Output

```
Sum: 75
```

Explanation of the above Program

Below is the explanation of the above program. With a description explaining the program's meaning and use.

Sections	Description
/ ***file:sum.c** *** author: you** *** description: program to find sum.** ***/**	It is the comment section and is part of the description section of the code.
#include<stdio.h>	Header file which is used for standard input-output. This is the preprocessor section.
#define X 20	This is the definition section. It allows the use of constant X in the code.
int sum(int y)	This is the Global declaration section includes the function declaration that can be used anywhere in the program.
int main()	main() is the first function that is executed in the C program.
{…}	These curly braces mark the beginning and end of the main function.
printf("Sum: %d", sum(y));	printf() function is used to print the sum on the screen.

Sections	Description
return 0;	We have used int as the return type so we have to return 0 which states that the given program is free from the error and it can be exited successfully.
int sum(int y) **{** **return y + X;** **}**	This is the subprogram section. It includes the user-defined functions that are called in the main() function.