STRUCTURE OF THE PROGRAM

The structure of a C program, foundational to the basic structure of C programming, is categorized into six key sections:

1. Documentation,

2. Link,

3. Definition,

4. Global Declaration,

5. Main() Function

6. Subprograms.

The Main() function is essential in every C program, highlighting what is required in each C program. This organization facilitates readability, modification, and documentation, providing a clear structure of C language with examples to illustrate the concepts.

Sections of the C Program

Documentation:

The documentation section consists of a set of comment lines giving the name of the program,the author, and other details which the compiler ignores.

In a C program,
Single-line comments can be written using two forward slashes i.e., //.
Multi-line comments can be written using /* */.

Example

```
/*
 * Author Name:
 * Date:
 * Program Name:
 */
```

Link Section :

All header files are included in this section.A header file is a file that consists of C declarations that can be used between different files. It helps us in using others' code in our files. A copy of these header files is inserted into your code before compilation.

Example

```
#include <stdio.h>
```

Definition:

A preprocessor directive in C is any statement that begins with the "#" symbol. The #define is a preprocessor compiler directive used to create constants. In simple terms, #define basically allows the macro definition, which allows the use of constants in our code.

Example #define MAX 100

We've created a constant BORN which is assigned a value of 2000. Generally, uppercase letters are preferred for defining the constants. The above constant BORN will be replaced by 2000 throughout our code wherever used.

#define is typically used to make a source program easy to modify and compile in different execution environments.

The define statement **does not** end with a semicolon.

Global Declaration :

This section includes all global variables, function declarations, and static variables. The variables declared in this section can be used anywhere in the program. They're accessible to all the functions of the program. Hence, they are called global variables**.**

Main Function

In the structure of a C program, this section contains the main function of the code. The C compiler starts execution from the main() function. It can use global variables, static variables, inbuilt functions, and user-defined functions. The return type of the main() function can be void and also not necessarily int.

Example

```
int main(void)

{

  printf("Hello world”);

}
```

Sub programs

This includes the user-defined functions called in the main() function. User-defined functions are generally written after the main() function irrespective of their order.

When the user-defined function is called from the main() function, the control of the program shifts to the called function, and when it encounters a return statement, it returns to the main() function.

Structure of C program with example

```
//Author Name:
//Date:
/* Name of the Program:          */

#include <stdio.h>
#include<conio.h>
#define MAX 100

int a=50;
void display();

void main()
{
   printf(“Hello……..I am inside the main function”);
   display();
   getch();
}

Void display()
{
  Printf(“Hello …………..I am inside he display function or user defined function”);
}
```