Learn CODE: Start with C

Article is prepared by –

***Sorowar Mahabub**, Studies B.Sc in CSE,*

*Int' Islamic University Chittagong (**IIUC**)*

***Cell:** 01834756433, 01521564157(whasApp available)*

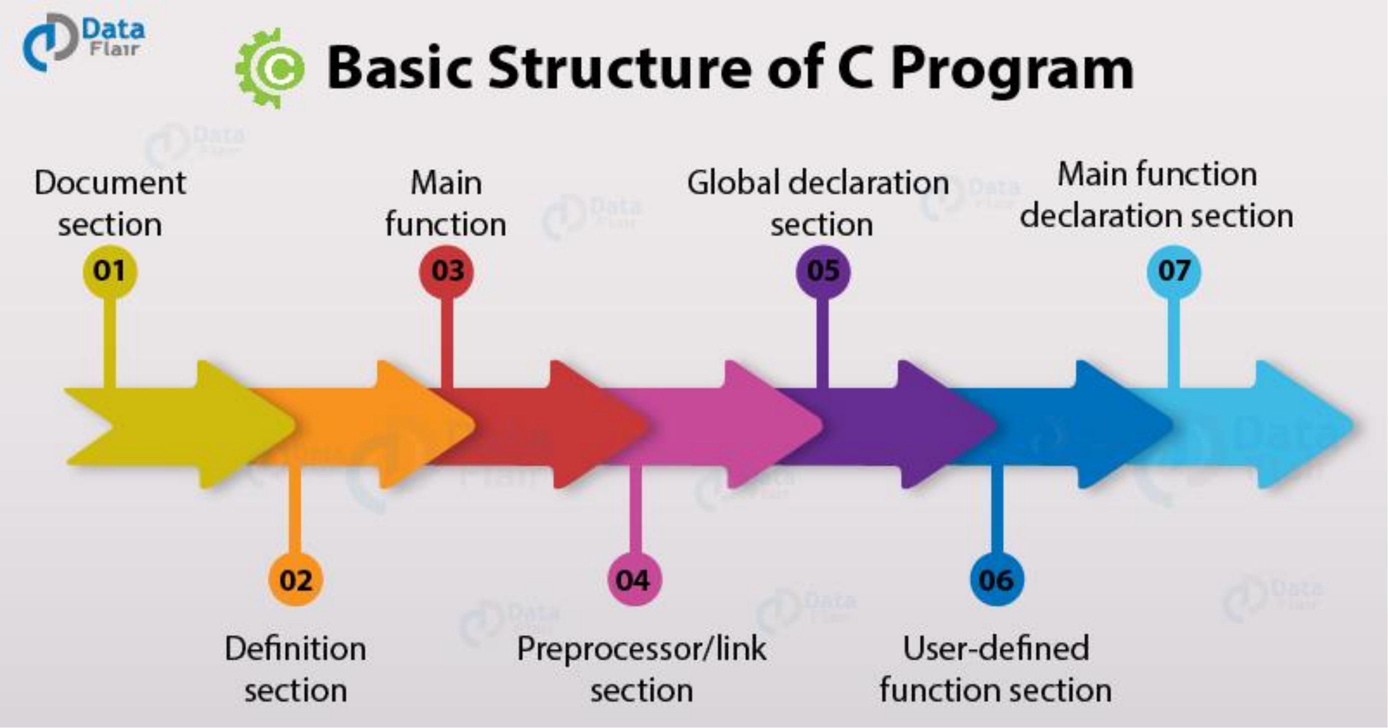
***Email :** sorowarmahabub1709vip@gmail.com*

Article Reference : Google, DataFlair_blog

Basic Structure of C Program (*Learn the in 7 Mins*)

Programming in C is a difficult task for someone who is completely oblivious to the basic structure of a C program. After completing this tutorial, you would learn how the Structure of C Program looks like and soon you would be comfortable writing your own programs with ease!

So, what are you waiting for? Let's start the Basic Structure of the C program.


Data Flair
Basic Structure of C Program
Document section
01
Definition section
02
Main function
03
Preprocessor/link section
04
Global declaration section
05
User-defined function section
06
Main function declaration section
07

(This Picture Collected from DataFlair)

Anatomy of a C Program

C language has a bundle of protocols that define its programming activities.

The C programming language came into existence when its developers were working on the development of the Unix operating system using the B language, out of which C evolved. The B language lacked certain features that led to the ***introduction of C.*** These features constituted the part of the C program upon which it was is built.

Parts of C program-

1. **# include <stdio.h>** – This command is a ***preprocessor directive in C*** that includes all standard input-output files before compiling any C program so as to make use of all those functions in our C program.
2. **int main()** – This is the line from where the *execution of the program starts*. The main() function starts the execution of any C program.

3. **{ (Opening bracket)** – This indicates the *beginning of any function* in the program (Here it indicates the beginning of the main function).
4. **/* some comments */** – Whatever is inside /*——-*/ are *not compiled and executed*; they are only written for user understanding or for making the program interactive by inserting a comment line. These are known as multiline comments. Single line comments are represented with the help of 2 forward slashes "//——".
5. **printf("Hello World")** –The printf() command is included in the C stdio.h library, which helps to *display the message on the output screen*.
6. **getch()** – This command helps to *hold the screen*.
7. **return 0** –This command terminates the C program and returns a null value, that is, 0.
8. **} (Closing brackets**)- This indicates the *end of the function*. (Here it indicates the end of the main function)

Example of C Program Structure

The "Hello World!" example is the most popular and basic program that will help you get started with programming. This program helps you display the output "Hello World" on the output screen.

With the help of this example, we can easily understand the basic structure of a C program.

```
#include <stdio.h>
int main()
{
// Our first basic program in C
printf("Hello World!\n\n");
return 0;
}
```

Key takeaway: \n is used to move the cursor to the new line.

You can practice the code explained above and compile it on your machine to get an essence of C programming.

Steps involved to get the desired output

Well, when you run a C program and get the output on your screen, there is a series of steps that come in the way. To get the desired output, you need to follow all the steps. These are the steps involved while writing a program in C.

1. Create
2. Compile
3. Execute or run
4. Desired output

First of all, try to code the program in the most precise manner following the protocols of C programming like,

- C is a case-sensitive programming language.
- **Each line of code in C ends with a semicolon(;), except the function definition.**

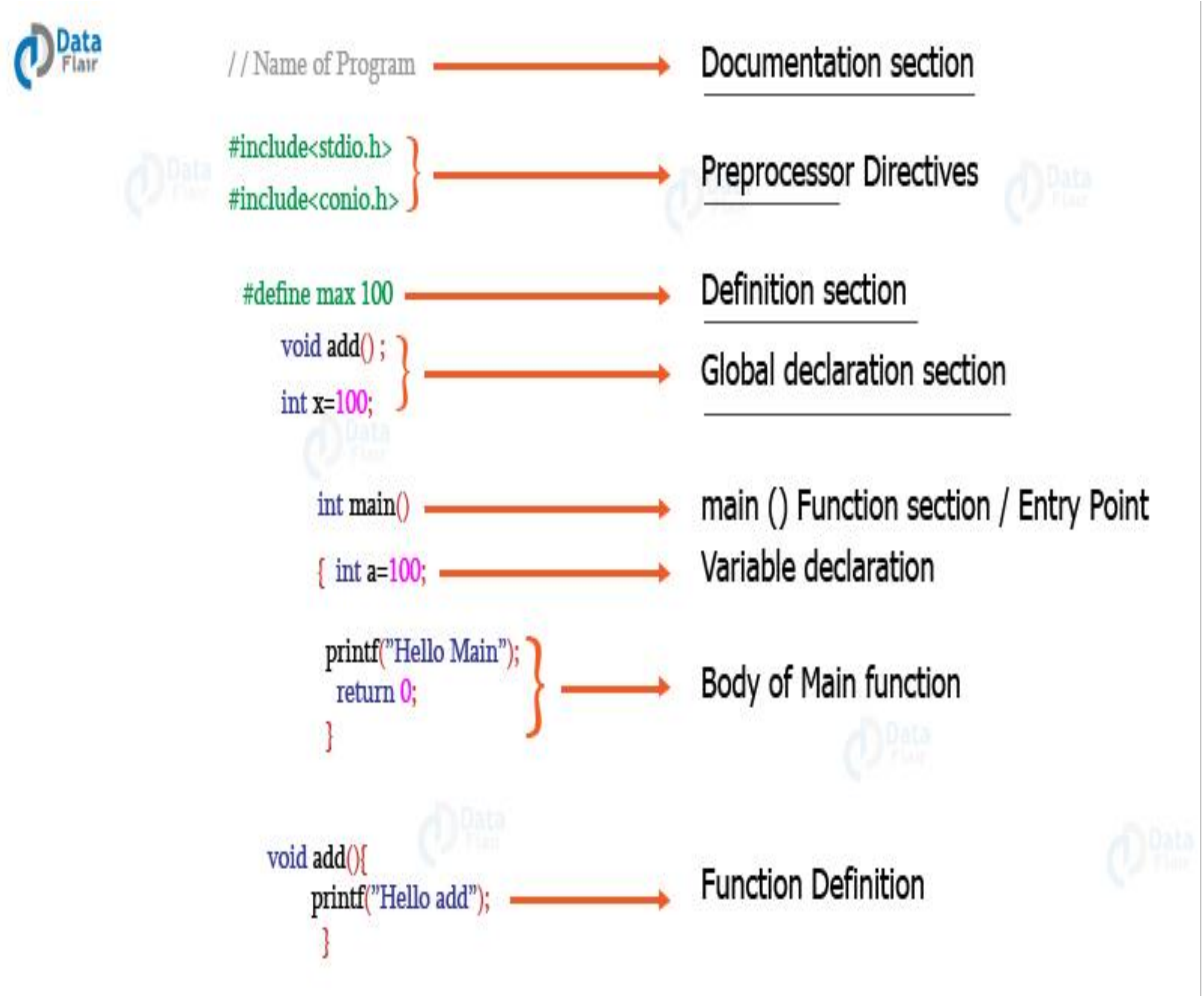
Also, if you don’t want to install the compilers on your system, you can also use the online compilers available to run your code and get the output.

Once, your code gets executed you will surely get the desired output on your output screen.

Basic Structure of C Program

The components of the basic structure of a C program consists of 7 parts

1. Document section
2. Preprocessor/link Section
3. Definition section
4. Global declaration section
5. Function declaration section
6. Main function
7. User-defined function section



```
//Name of Program
#include<stdio.h>
#include<conio.h>
#define max 100
  void add() ;
  int x=100;
    int main()
    { int a=100;
     printf("Hello Main");
      return 0;
     }
  void add(){
      printf("Hello add");
     }
```

This Picture Collected from DataFlair)

Still, not getting? Let’s discuss every basic component of the C program with the help of an example.

```
/*
Documentation section
C programming structure
Author: Sorowar Mahabub, Learn CODE: Start with C!
*/
#include <stdio.h> /* Link section */
int subtract = 0; /* Global declaration, definition section */
int all (int, int); /* Function declaration section */
int main () /* Main function */
{

printf("Welcome to Learn CODE: Start with C!\n\n");

printf ("This is a C program \n");
subtract= all (25,10);
printf ("Subtraction of the two numbers : %d \n", subtract);
return 0;
}
int all (int x, int y) /* User defined function */
{
return x-y; /* definition section */
}
```

1. Documentation Section

It is the section in which you can give comments to make the program more interactive. The compiler won’t compile this and hence this portion would not be displyed on the output screen.

2. Preprocessor directives Section

This section involves the use of header files that are to included necessarily program.

3. Definition section

This section involves the *variable definition and declaration in C*.

4. Global declaration Section

This section is used to define the global variables to be used in the programs, that means you can use these variables throughout the program.

5. Function prototype declaration section

This section gives the information about a function that includes, the data type or the return type, the parameters passed or the arguments.

6. Main function

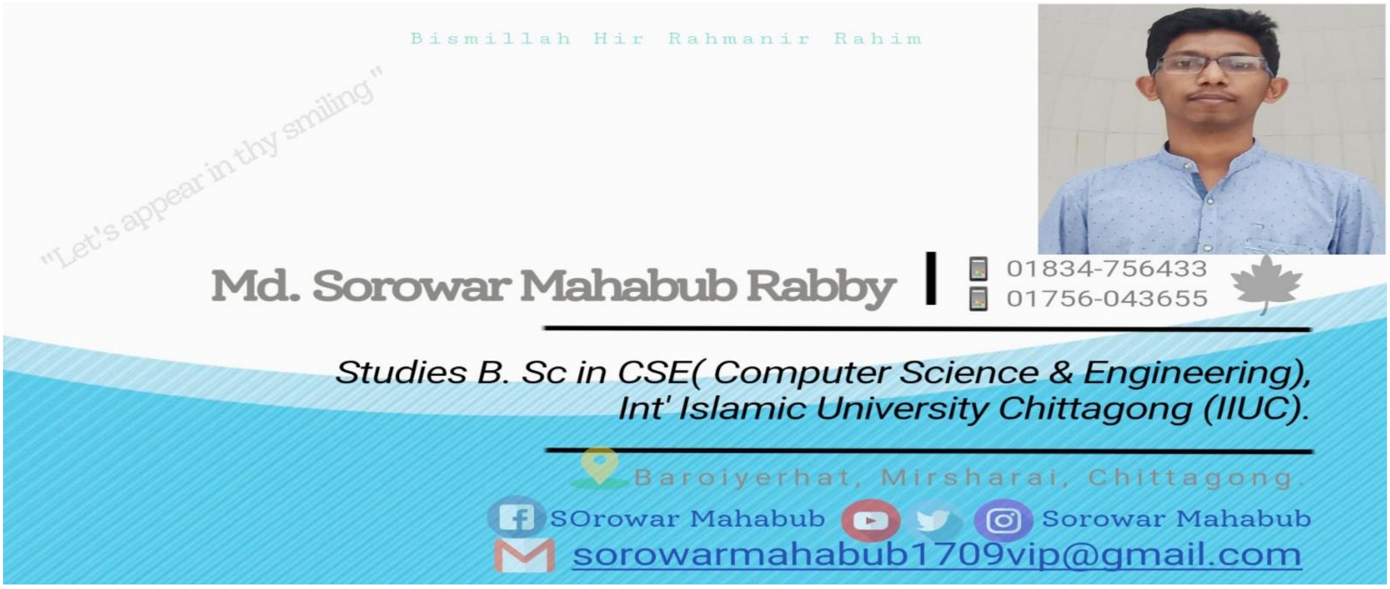
It is the major section from where the execution of the program begins. The main section involves the declaration and executable section.

7. User-defined function section

When you want to define your function that fulfills a particular requirement, you can define them in this section.

Summary

We learned about the basic structure of a C program and also the commands necessary to build a C program. Every C program involves many sections that complete the program; we have gone through all these sections with the help of some examples.



Bismillah Hir Rahmanir Rahim

"Let's appear in thy smiling"

Md. Sorowar Mahabub Rabby

01834-756433
01756-043655

Studies B. Sc in CSE(Computer Science & Engineering),
Int' Islamic University Chittagong (IIUC).

Baroiyerhat, Mirsharai, Chittagong.

SOrowar Mahabub

Sorowar Mahabub

sorowarmahabub1709vip@gmail.com