Lesson Number: 6

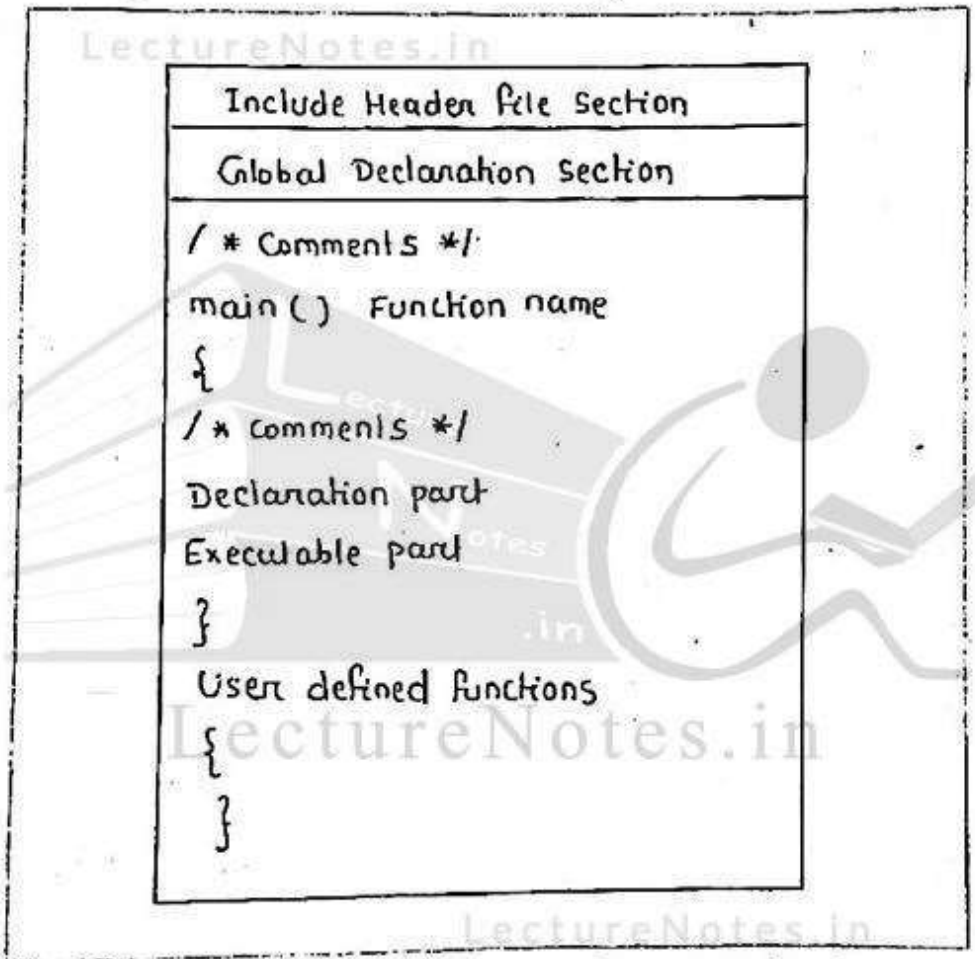
Structure of a C program :

→ Every C program contains a number of several building blocks known as functions.

→ Each function of it performs task independently.

→ A function is subroutine that may consist of one or more statement.

→ A C program comprises the following sections.



(Structure of a 'C' program)

a) Include header file section :

→ The header files or preprocessor directives gives instructions to the compiler to include compiler options (# include).

→ C programs depends upon some header files for function definition that are used in program. Each header file by default

[Reference : programming in ANSI, E. Balaguruswamy]

is extended with .h .

→ The file should be included using #include directive as given below.

Example : #include <stdio.h> or #include "stdio.h"

→ #include <stdio.h> tells the Compiler to include information about the standard input/output library. This file is also Compiled with original program.

Name of the Function	Description
getchar()	Reads a character from the keyboard ; waits for carriage return.
getch()	Reads the character without an echo to the screen.
getche()	Same as getch(), but echoes the character.
putchar()	writes a character to the screen.
putch()	writes a character to the screen.
gets()	Reads a string of characters.
puts()	writes a string to the screen.
printf()	writes data in various formats to output device.
Scanf()	Reads data in various formats from input device.

(Standard input output library functions)

b) Global declaration :

→ Variables declared outside main() are called global variables and they can be used in the main program block and sub program block.

c) Function main :

→ The main() is a special function used by the C system to tell the Computer where the program starts.

→ Every program must have exactly one main function.

→ If we use more than one main function, the Compiler cannot tell which one marcks the begining of the program.

→ C permits different forms of main statements.

- main()
- int main()
- void main()
- main (void)
- void main (void)
- int main (void)

The empty pair of parentheses indicates that the function has no arguments. This may be explicilly indicated by using the keyword void inside the parantheses. We may also specify the keyword int or void before the word main. The keyword void means that the function does not return any information to the operating system and int means that the function returns an integer value to the operating system. when int is specified the last statement in the program must be "return 0".

→ The program execution starts with the opening brace ({) and ends with the closing brace (}). Between these two braces the programmer should declare the declaration and executable part.

d) Declaration part :

→ The declaration part declares the entire variables that are used in executable part.

→ The initialization of variables are also done in this section. This initialization means providing initial value to the variables.

→ Variables declared inside main() are called local variables, and they are used only in block in which they are declared. Subprograms / functions can also have local variables.

e) Executable part :

→ This part contains the statements following the declaration of the variables.

→ This part contains a set of statements or a single statement. These statements are enclosed between the braces.

f) User defined function :

→ The function declared by the user are called user-defined functions.

→ These functions are generally defined after the main() function.

→ They can also be defined before main() function. This portion is not compulsory.

g) Comments :

→ Comments are not necessary in the program.

→ However, to understand the flow of program, the programmer can include comments in the program.

→ Comments are to be inserted by the programmer. It is useful for documentation.

→ The clarity of the program can be followed if it is properly documented.

→ Comments are nothing but some kind of statements which are placed between the delimiters /* & */. The compiler does not execute comments. Thus we can say that comments are not the part of executable programs.

→ The user can frequently use any number of comments that can be placed anywhere in the program. The comments and statements can be nested.

→ The user should select the OPTION MENU of the editor and select the COMPILER - SOURCE - NESTED COMMENTS ON/OFF).

→ The comments can be inserted with a single statement or in nested statements.

Examples:

- /* This is single comment */
- /* This is an example of /* nested comments */ */
- /* This is an example of
  comments in
  multiple lines */ /* It can be nested */

Header Files :

→ Library functions are grouped category-wise and stored in different files known as header files..

→ C programs depends upon some header files for function definition that are used in program . Each header file by default is extended with .h .

→ The header files should be included. using #include directive as given below .

Example : #include <stdio.h> or #include "stdio.h"

1) #include <stdio.h>

→ The #include <stdio.h> tells the compiler to include information about the standard input output library function .

→ The standard input output library function include puts(), gets(), printf(), scanf() etc .

2) #include <math.h>

→ The #include <math.h> tells the compiler to include information about the standard mathematical functions.

→ The standard mathematical function include cos(d), exp(d), fabs() etc .

3) #include <ctype.h>

→ The #include <ctype.h> tell the compiler to include information about the character testing and conversion functions .

→ The standard character testing and conversion functions include isalnum(c), isalpha(), isdigit() etc .

4) #include <stdlib.h>

→ The #include <stdlib.h> tells the compiler to include the information about the utility functions such as string conversion routines, memory allocation routines, random number generator etc.

→ The standard stdlib.h functions include abs(i), atof(s), atoi(s), atol(s) etc.

5) #include <string.h>

→ The #include <string.h> tells the compiler to include the information about the string manipulation function.

→ The standard string manipulation function include strcmp(S1, S2), strcpy(S1, S2), strlen(s) etc.

6) #include <time.h>

→ The #include <time.h> tells the compiler to include information about the time manipulation functions.

→ The standard time manipulation function include time(P), difftime(t1, t2) etc.

7) #include <graphics.h>

→ The #include <graphics.h> tells the compiler to include information about the graphical functions.

→ The standard graphical function include drawscreen(), gotoxy() etc.

8) #include <conio.h>

→ The #include <conio.h> tells the compiler to include console input output functions.

→ The standard console input output function include clrscr().

Assignment : what is a header file ?

→ Library functions are grouped category-wise and stored in different files known as header files.

→ The header files or preprocessor directives gives instructions to the compiler to include compiler options (#include).

→ C programs depends upon some header files for function definition that are used in program. Each header file by default is extended with .h .

→ The file should be included using #include directive as given below :

Example : #include <stdio.h> or #include "stdio.h"

→ The #include<stdio.h> tells the compiler to include information about the standard input/output library. This file is also compiled with original program.

Assignment : what is a global variable ?

Soln variables declared outside main() are called global variable and they can be used in the main program block and sub program block .

Assignment : Write short notes on Comment line in C program .

→ Comments are not necessary in the program . To understand the flow of program, the programmer can include comments in the pgm.

→ Comments are to be inserted by the programmer within the delimeter /* and */.

→ Comments are not the part of executable programs .

→ The user can frequently use any no. of comments, anywhere in the pgm.

→ Example : /* This is Single Comment */