

# COMPILING A SIMPLE C PROGRAM - I

This chapter covers some simple C programs which will help the student to understand the structure of a C program. The student is encouraged to try as many programs as possible to become familiar with the syntax of the language and the types of errors encountered.

C is a case sensitive language. This means that lowercase and uppercase letters are treated differently. So the C compiler interprets a variable named "VAR" as different from a variable named "var" or "Var". Also unlike other languages like Fortran, in C it does not matter where you start typing your code (indentation). But it is advisable to follow some coding conventions in order to develop programs which are readable and hence easily maintainable.

Consider the following program :

```
/* Program to display a string on the screen */  
  
#include <stdio.h>  
  
void main()  
{  
    printf("Hello World \n");  
}
```

Let us first understand the different parts of this C program. This program begins with a comment enclosed between `/*` and `*/`. A comment is a note or

explanation that you put into your source code. Comments are used primarily to document the meaning and purpose of your source code, so that you can remember later how it functions and how to use it. You can also use a comment to temporarily remove a line of code. Simply surround the lines with the comment symbols. Comments cannot be nested, but can span several lines. Generally they are placed at the beginning of a program or function or they may co-exist on lines that have other code. Comments make a program readable and maintainable.

The second line of the program is a preprocessing directive. Preprocessors are discussed in subsequent chapters (see Chapter 14). This preprocessing directive tells the compiler to include the `stdio.h` header file. This is a standard header file containing the functions and constants used for console and file input and output. If the header file is supplied by the compiler system then enclose it with angle brackets (`< >`), else if it is your own include file then surround it with double quotes (`" "`). More explanation can be found in reference material on standard C Library files.

`main()` is a special keyword that indicates to the C system the entry point of a program. This is where the program is to begin execution. The keyword before `main()` indicates the type of the return parameter of the function. In this case `void` indicates that the function `main()` does not return a value. Input parameters to the main function are enclosed in the parentheses after `"main"`. In this case `main()` indicate that no parameters or arguments are passed on to

this function. The syntax of a function and its working will be explained in detail in Chapter 17 on Functions.

Following main, is the body of the program which consists of several statements enclosed within a pair of curly braces. In the above program we have only one statement which is the printf() routine. This is a C function that prints its arguments at the terminal or on the video display. This statement prints the statement "Hello World" on to the screen. The characters backslash (\) and the letter n constitute the newline character. This tells the C system to print the statement and proceed to a new line. This is similar to the carriage return key on a typewriter.

Note: All program statements in C must be terminated by a semicolon.

The following program illustrates the usage of \n for building an output formed by two rows.

```
/* Program for displaying two one string in two rows */  
  
#include <stdio.h>  
  
void main()  
{  
    printf("Hello World \n Programming in C is interesting  
! \n");  
}
```