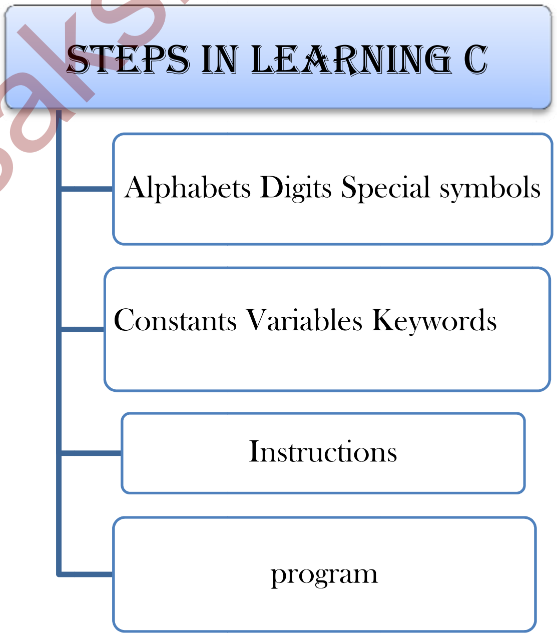
INTRODUCTION TO C- PROGRAMMING

Introduction

R.Sagar, Sr Software Engineer
SRM Software Labs

There is a close analogy between learning English and learning c-language. The classical method of learning English is to first learn the alphabets used in the language, then learn to combine these alphabets to form words, which in turn are combined to form sentences and sentences are combined to form paragraphs. Learning C is similar and easier. Instead of straight-away learning how to write programs, we must first know what alphabets, numbers and special symbols are used in C, then how using them as constants, variables and keywords and finally how are these combined to form an instruction. A group of instructions would be combined later on to form a program.

Steps in learning C



STEPS IN LEARNING C

Alphabets Digits Special symbols

Constants Variables Keywords

Instructions

program

C is developed from two previous languages, known as BCPL and B. BCPL was developed by Martin Richards in 1967. These languages are “type less” languages i.e. every data item in program occupy one “word” in memory. C is a general purpose programming language developed **by Dennis M. Ritchie** at bell laboratories and it was originally implemented on a DEC PDP-11 computer in 1972. The C language imports many important features of BCPL and B and added other features like data typing. Initially C is used to develop the UNIX operating system. C is hardware independent programming language. C language also called as high level and structured programming language.

WHY TO USE C

- The Portability Of The Compiler;
- The Standard Library Concept;
- A Powerful And Varied Of Operators;
- An Elegant Syntax;
- Ready Access To The Hardware When Needed;
- And the ease with which applications can be optimized by hand-coding isolated procedures

C is often called a "Middle Level" programming language. This is not a reflection on its lack of programming power but more a reflection on its capability to access the system's low level functions. Most high-level languages (e.g. FORTRAN) provide everything the programmer might want to do already build into the language. A low level language (e.g. assembler) provides nothing other than access to the machines basic instruction set. A middle level language, such as C, probably doesn't supply all the constructs found in high-languages - but it provides you with all the building blocks that you will need to produce the results you want.

APPLICATIONS OF C

C was initially used for system development work, in particular the programs that make-up the operating system. Why use C? Mainly because it produces code that runs nearly as fast as code written in assembly language. The application of C might be,

- Operating Systems
- Language Compilers
- Assemblers
- Text Editors
- Network Drivers
- Modern Programs
- Data Bases
- Language Interpreters
- C language is used for creating **computer applications**
- Used in writing **Embedded software's**
- Firmware for various electronics, industrial and communications products which use micro-controllers.
- It is also used in developing **verification software, test code, simulators** etc. for various applications and hardware products.
- UNIX kernel is completely developed in C Language.

FEATURES OF C LANGUAGE

- General purpose structured programming language.

- **Portability**

  - C Programs are portable i.e. they can be run on any Compiler with Little or no Modification. Compiler and Preprocessor make it Possible for C Program to run it on Different PC.

- **Powerful**

  Provides wide verity of **Data Types, Functions and useful Control & Loop Control Statements.**

  - **Middle level language.**
  - **Easy to debug.**

- **Efficient Use of Pointers**

  - C Supports efficient use of pointer. Pointers have direct access to memory.

- **Modular Programming**

  - **Modular programming** is a software design technique that increases the extent to which software is composed of separate parts, called **modules.**
  - C Program Consist of Different Modules that are integrated together to form complete program.

STRUCTURE OF A C-PROGRAM

The structure of the c-program means some set of rules are formed to make a program to solve. So there is a structure for the c-program to write.

The structure of the c-program is,

```
Documentation

Preprocessor directives

Global declarations

main ()

{

        Local declarations

        Program statements

        Calling user defined functions (user optional)

}

User defined functions. (User optional)
```

Documentation

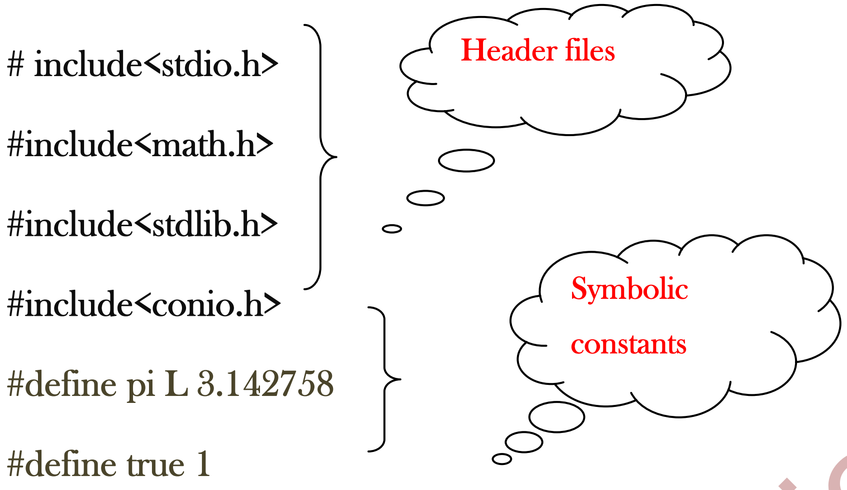
To increase the readability of the program documentation is necessary. In this documentation comment lines are included.

Example: -// Program to find out biggest of 2 numbers a and b.

Preprocessor Directives

A statement which is preceded by the # symbol is called the Preprocessor Directive. This statement instructs the compiler to include the preprocessor files into the program.

For example the C in-built statement printf we used in this program comes under stdio.h header files.



```
# include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<conio.h>
#define pi L 3.142758
#define true 1
```

Global declarations

The variables and the user defined functions are declared outside the main function.

Local declarations

Those variables declared inside the main function. Then it is called the local declaration.

User defined function

The sub program written by the user to make the desired task executed. That program is user defined program. That program contains set of statements.

This is optional for the user.

EXECUTING A C PROGRAM

Writing a c-program involves series of steps,

- Step 1: Creating a program.
- Step 2: Compiling the program.
- Step3: Linking the program with functions that are needed from the c- library.
- Step 4: Executing the program.

- Although all the above steps remain same irrespective of the operating system, system commands for implementing the steps and conventions for naming files may differ on the different systems.

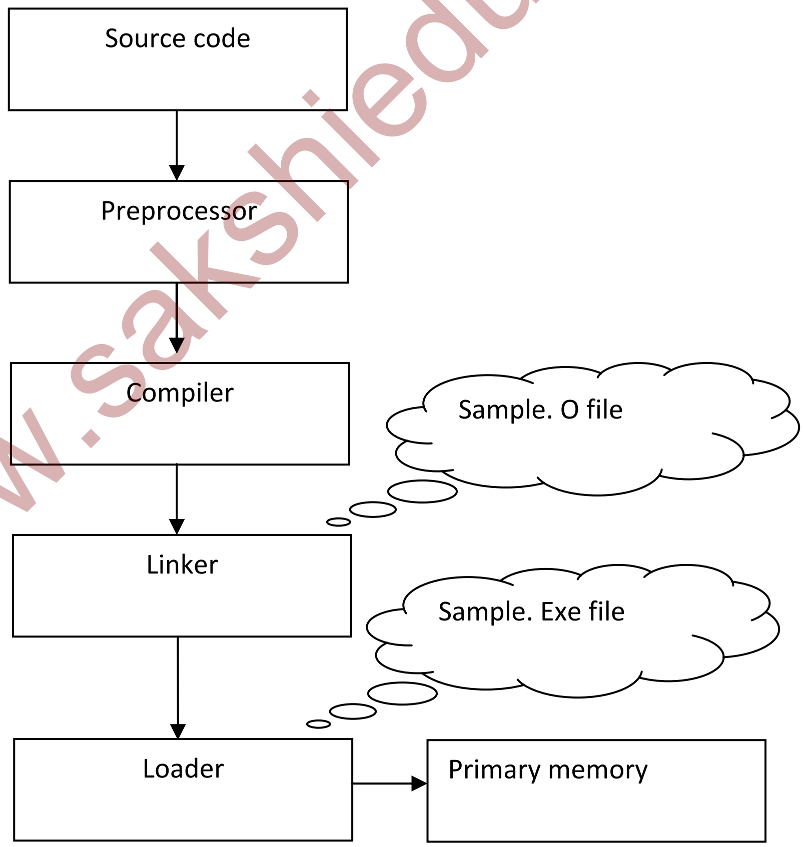


Figure: - Flow chart shows the sequence of execution of the program

- The "Compiler" [it is a System Software], will help us to convert our high level program (Source code) to the low level program (object file).
- When you will compile your program that will generate some kind of Object file (machine understandable code), if and only if your program do not have any kind of error in the program otherwise it will point out those.

- Now we will talk about the "Linker", when you add some of the library files within your program, and use some of those functionality at that time to execute that code, first your C program needs to linked up with those files , to use those functions and that it will copy the whole code of that file and put it within your C program, and will help you to run your program without any error, that is the function of the Linker, and one more thing Linker will also combine the Object file and the Library Files in to the thing called the .exe (which is an Executable file) or we can say that a file that is portable, means we do not need to use the source code each and every time to deal with that program(it will be platform dependent). At the end there is one more thing which is known as the Loader, which will help your program to get the sufficient memory of the RAM, so ultimately it will help you to load your program on to RAM and now when you are going to run that program, actually System Software [Loader- which is system software] will execute that particular executable file. This is how the 'C' program is going to be executed.

TEST

1. Which of the following language is predecessor to C Programming Language?

   a. A

   b. B

   c. C++

   d. BCPL

   Answer: d

2. C was developed in the year ?

   a. 1970

   b. 1972

   c 1976

   e. 1980

   Answers:-b

3. C programming language was developed by ?

   a. Dennis Ritchie.

   b. Ken Thompson

   c. Bill Gates.

   d. Peter Norton.

   Answer: - a

4. C-language is available for which of the following operating systems?

a. DOS

b. Windows

c. UNIX

d. All of these

Answers :- d

5. C is a ------------language

a. High level language. Middle level language.

b. Low level language.

c. Machine level language

d. Middle level language.

Answer: - d

6. Which symbol is used as terminator in C

a. ;

b. #

c. ~

d. |

Answer: - a

7. Which of the following is used as pre-processor statement?

a. !

b. ~

c. #

d. &

Answer : c

8. The program transfers the executable image from disk to memory.

a. Linker.

b. Compiler.

c. Assembler.

d. None of the above.

Answer : a