



## CHAPTER-1: Introduction

1. About program and programming
2. A sample program in “C” language
3. Editing, compiling, linking, and execution of a ‘C’ Program

### About program and programming

**Programming** a computer means preparing a set of instructions or commands or orders for it to follow. These instructions can be written in one of the several **languages like ASSEMBLY, BASIC, COBOL, FORTRAN, PASCAL, C, C++, VB, JAVA, etc.** Eventually the instructions need to be translated in the case of middle-level language or high-level languages and assembled in case of assembly language to produce low level or machine level code or binary instructions. Because it is the translated version, which are directly understand and executed by the CPU.

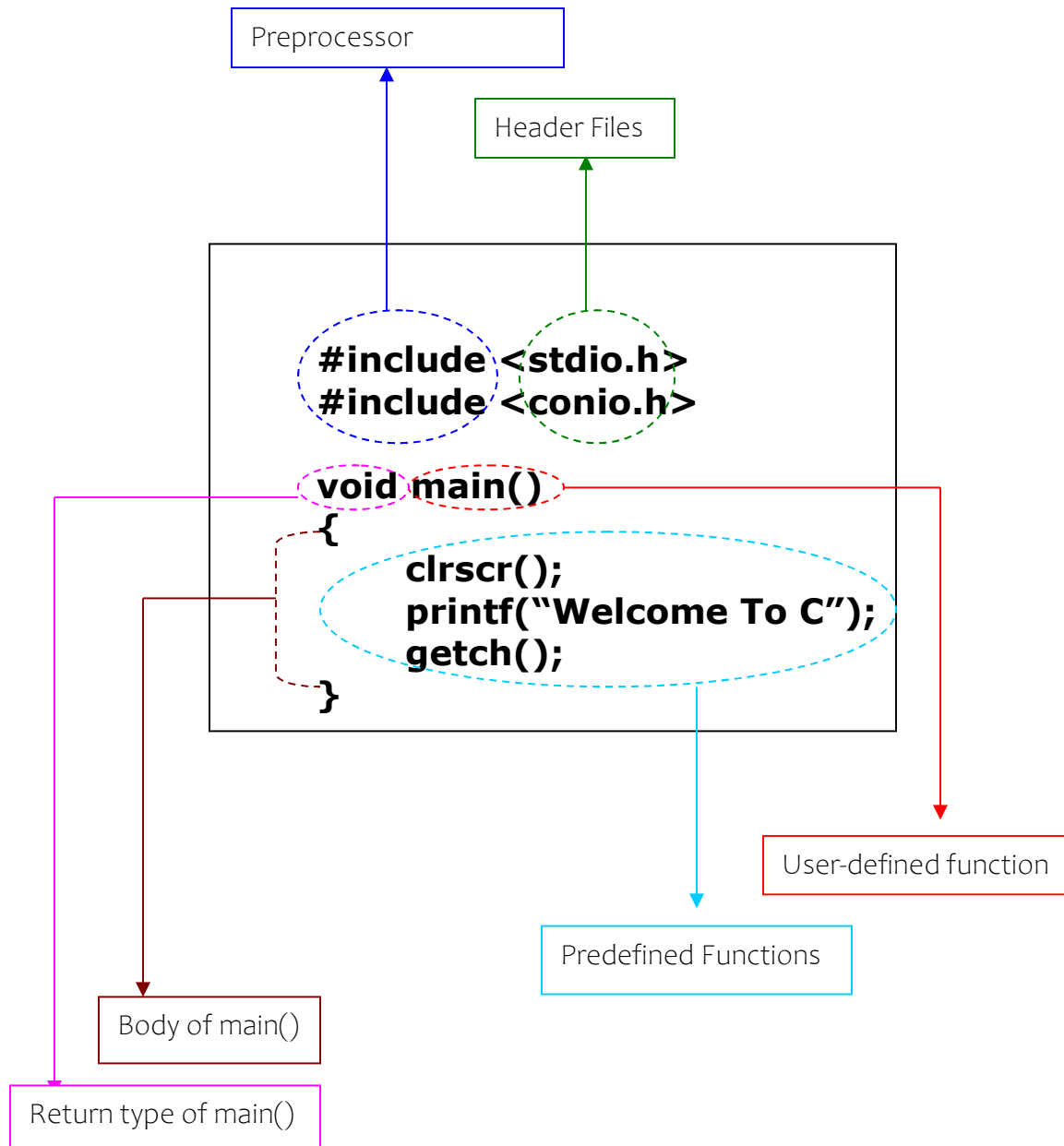
A **command** or **order** given to a computer is referred to as an **instruction**. It normally consists of a code to indicate the operation to be performed and addresses in memory where the operands would be found. A set of sequenced instructions is referred to as a **program**, which is developed or designed to direct and control the operations of a computer to solve a specific problem or task or goal. The person who designs, writes, tests, and maintains a program, is known as a **programmer** and process of development of a program is referred to as **programming**.

**Programming is a skill**, just as bicycling, swimming or driving are skills and it can be learned only by practice. Just as no amount of poolside theorizing can teach us swimming: if we wish to swim, at some point we will have to jump into the water; so also reading alone can never suffice to make us into a proficient programmer. **The only way to become a good programmer is a practice, practice, practice and still more practice...**

Wish You, Happy Programming!  
Welcome to the **Team MyR&D**



## A sample program of ‘C’ language



**Fig: - Showing a sample program of ‘C’ language**



**Important Points: -**

1. **void** is a reserved keyword and indicates return type of the function main, which tells the compiler that nothing is to return to its caller (os).
2. opening curly brace { indicates the beginning of the function main and close curly brace } indicates the end of the function main.

A program written in ‘C’ language is referred to as a **c-program** or a c-source code or a **c-source file**. A c-source file must follow the following properties: -

- Secondary name of the file (i.e. extension) must be dot c (.c)
- It must follow the **syntax** and **reserved keywords** supplied by ‘C’

**Example: -**

**SAMPLE1.C**

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    printf("Welcome to the Team MyR&D");
    getch();
}
```

OR,

**SAMPLE2.C**

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr(),printf("Welcome to C"),getch();
}
```



### Important Note: -

- ‘C’ is a case-sensitive language and uses the small case letters only

### Syntactical Rule: -

- A statement in ‘C’ must be terminated by a semicolon (;)
- Two things in ‘C’ can be separated by a comma (,)
- Header files are optional in ‘C’

## Editing, Compiling, Linking and Execution of a ‘C’ Program

- **Editing:** - Creating a ‘C’ source file with extension “.c”
- **Compiling:** - Translating source file into object file (“.obj”)
- **Linking:** - Adding library objects to the object file generated from source file and generating an executable file (“.exe”) or load module, which can be executed.
- **Execution:** - Loading the executable file (“.exe”) into the main memory and executing the program.

### Editing: - Creating A ‘C’ Source File

‘C’ **source file** is a textual file. We can use any **Text Editor** or an **IDE** (Integrated Development Environment) to create a “C” source file. Following are the examples: -

- DOS (command.com) – **Terminal**
- DOS Editor (edit.com) - **Editor**
- Notepad (notepad.exe) - **Editor**
- Notepad++ - **Editor**
- TurboC (tc.exe) - **IDE**
- Visual Studio Code - **IDE**
- Online REPL (<https://repl.it>) - **IDE**

**NOTE:** Any of the above Editor/IDE you can use to write your ‘C’ program. After writing your program, you will have to compile and then link with the library object to get the actual executable file (“.exe”). For compiling a ‘C’ program, you will have to have any ‘C’ compiler and for linking you will need to have linker. If you are using any IDE, you will get embedded options in it to compile, link and run your program. But I will suggest you to start with simple editor instead of using IDE to feel the actual compilation and linking process.



**Examples created using different options:**

**Example-1:** Created using DOS Terminal (command.com)

```
C:\MEGASOFT>copy con SAMPLE.C
#include<stdio.h>
#include<conio.h>

void main()
{
    clrscr();
    printf("Welcome To C");
    getch();
}
^Z
1 file(s) copied.
C:\MEGASOFT>_
```

**Example-2:** Created using DOS Editor (edit.com)

```
C:\WINDOWS\system32\EDIT.COM
File Edit Search View Options Help
c:\MEGASOFT\SAMPLE.C

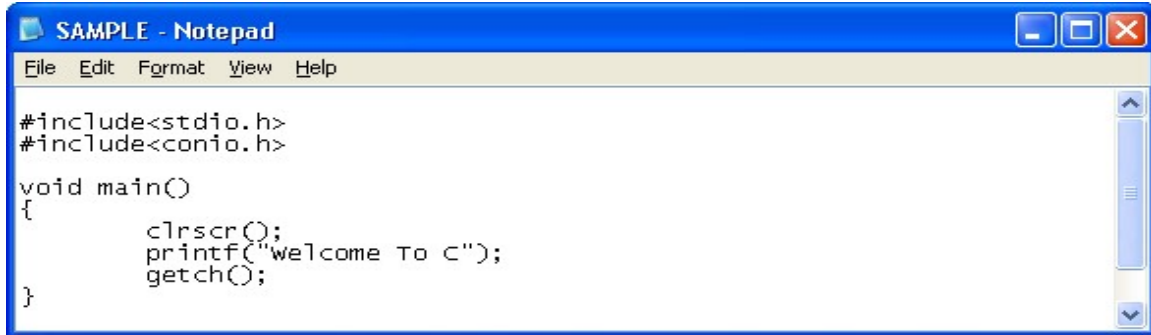
#include<stdio.h>
#include<conio.h>

void main()
{
    clrscr();
    printf("Welcome To C");
    getch();
}

F1=Help | Line:10 Col:1
```



### Example-3: Created using Notepad

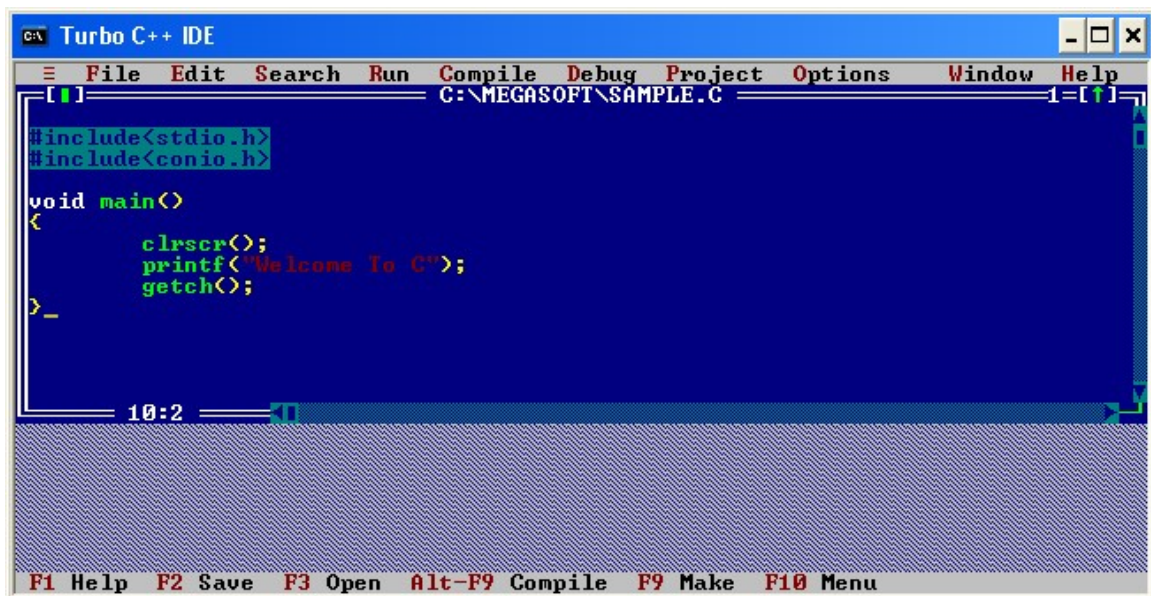


```
SAMPLE - Notepad
File Edit Format View Help

#include<stdio.h>
#include<conio.h>

void main()
{
    clrscr();
    printf("welcome To C");
    getch();
}
```

### Example-4: Created using TurboC (tc.exe)

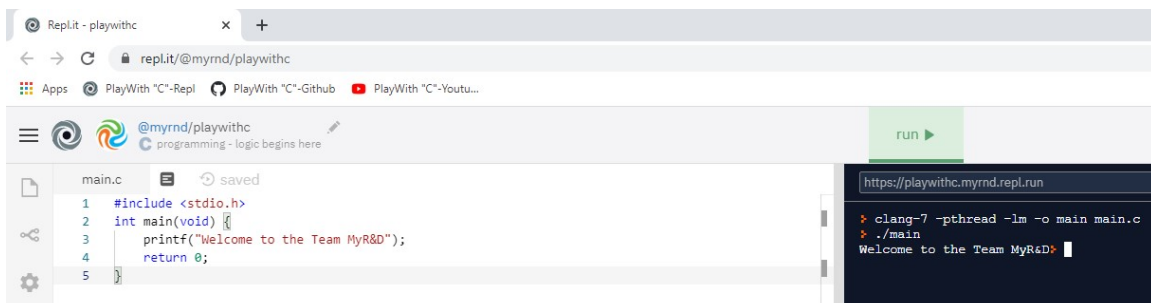


```
Turbo C++ IDE
File Edit Search Run Compile Debug Project Options Window Help
C:\MEGASOFT\SAMPLE.C

#include<stdio.h>
#include<conio.h>

void main()
{
    clrscr();
    printf("Welcome To C");
    getch();
}
```

### Example-5: Created using online REPL



```
Replit - playwithc
replit/@myrnd/playwithc
Apps PlayWith "C"-Repl PlayWith "C"-Github PlayWith "C"-Youtu...

@myrnd/playwithc
programming - logic begins here

main.c
1 #include <stdio.h>
2 int main(void) {
3     printf("Welcome to the Team MyR&D");
4     return 0;
5 }

run ▶

https://playwithc.myrnd.repl.run
> clang-7 -pthread -lm -o main main.c
> ./main
Welcome to the Team MyR&D
```



## Compiling: - Translating the source code (“.c”) into the object file (“.obj”)

The contents/notations/instructions/statements written in your ‘C’ program file don’t directly understand by the machine. Only the binary instructions are directly understood and executed by the CPU. Therefore, the codes written the “.c” file must be translated into its equivalent machine instructions.

Translating the source code into its equivalent machine code is known to as compiling. The program or software package, which performs compilation, is known as compiler. There are several compilers available from different vendors that you can use to compile your source code. For example:

- **Borland TurboC Compiler (tcc.exe)** – Widely used
- **Microsoft C**

Compiler takes the source file as input and gives object file as output:



Example: -



**Note:** - An object file (“.obj”) is nearly equivalent to machine instructions.

How to compile using “tcc.exe” on DOS Terminal: **tcc -I. <program.c>**

```
ex C:\WINDOWS\system32\cmd.exe - tcc -I. demo.c

C:\DEV\TURBOC3>dir demo.*
Volume in drive C has no label.
Volume Serial Number is C096-D745

Directory of C:\dev\Turboc3
05/10/2020  05:38 PM                56 demo.c
               1 File(s)                56 bytes
               0 Dir(s)  7,065,333,760 bytes free

C:\DEV\TURBOC3>tcc -I. demo.c
Turbo C++ Version 3.00 Copyright (c) 1992 Borland International
demo.c:
Turbo Link Version 5.0 Copyright (c) 1992 Borland International

Available memory 4152080

C:\DEV\TURBOC3>dir demo.*
Volume in drive C has no label.
Volume Serial Number is C096-D745

Directory of C:\dev\Turboc3
05/10/2020  05:38 PM                56 demo.c
05/10/2020  09:21 PM           6,440 DEMO.EXE
05/10/2020  09:21 PM           317 DEMO.OBJ
               3 File(s)           6,813 bytes
               0 Dir(s)  7,065,325,568 bytes free

C:\DEV\TURBOC3>
```



## How to compile in TurboC IDE

### Steps: -

1. Type/Open the Source File
2. Go to the **COMPILE** menu
3. Select the option **COMPILE**

**Note:** You can also use the shortcut: - **ALT+F9**

## To Confirm the creation of Object File

### Steps: -

1. Go to the **FILE** menu
2. Select the option **DOS Shell**
3. Type the command **DIR** then press **<Enter>** key

## Linking: -Adding the Library Object Codes with created Object File

A 'C' program uses the predefined or inbuilt library functions, whose object codes are far kept in the library. Hence that object codes must be merged/linked with the object file.

To add/link the object code of the functions with the object file to generate the executable file or load module is referred to as linking. And the program or software package, which performs the linking, is known as linker.

Linker takes “.obj” object file as input and gives “.exe” executable file as output.



### Example: -







## How to link Library Object Codes with created Object File

### Steps: -

1. Go to the **COMPILE** menu
2. Select the option **LINK**

### To Confirm the Creation of Executable File

### Steps: -

3. Go to the **FILE** menu
4. Select the option **DOS Shell**
5. Type the command **DIR** then press <Enter> key

## Execution: -Loading the Load Module or Executable File Into Main Memory

It is necessary to send/load the executable file or load module into the main memory so that execution of the instruction can take place is referred to as execution of a 'C' program.

To Execute the program, you can use the shortcut: - **CTRL+F9**

### How to link using “tlink.exe” on DOS Terminal:

Syntax: **tcc -L. <program.obj>**

```
C:\WINDOWS\system32\cmd.exe - tcc -l. demo.c
Volume Serial Number is C096-D745

Directory of C:\DEV\TURBOC3
05/10/2020  05:38 PM                56 demo.c
05/10/2020  09:46 PM            317 DEMO.OBJ
               2 File(s)            373 bytes
               0 Dir(s)  7,055,740,928 bytes free

C:\DEV\TURBOC3>tcc -L. demo.obj
Turbo C++ Version 3.00 Copyright (c) 1992 Borland International
Turbo Link Version 5.0 Copyright (c) 1992 Borland International

    Available memory 4193848

C:\DEV\TURBOC3>dir demo.*
Volume in drive C has no label.
Volume Serial Number is C096-D745

Directory of C:\DEV\TURBOC3
05/10/2020  05:38 PM                56 demo.c
05/10/2020  09:47 PM            6,440 DEMO.EXE
05/10/2020  09:46 PM            317 DEMO.OBJ
               3 File(s)            6,813 bytes
               0 Dir(s)  7,055,732,736 bytes free

C:\DEV\TURBOC3>
```