**Laboratory 1: Cover Sheet**

Name m\_ \_ Roll.No. \_

Date \_\_ Semester \_ Section \_

# Use the Borland Turbo C Environment and Program Structure.

|  |  |  |
| --- | --- | --- |
| **Activities** | **Remarks** | **Signature** |
|  |  |  |

Lab 01: Use the Borland Turbo C Environment and Program Structure.

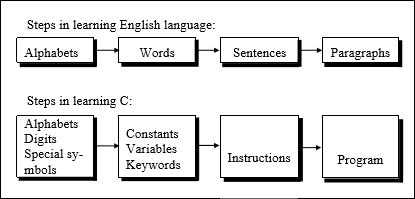
## Lab Objectives:

* Learn how to open the Borland Turbo C/C++, and use it
* How to write and execute a simple program that print a simple statement.
* Applying the escape characters like \n, \t,\\, etc.
* Using the single line and multiline comments.

## Background:

Communicating with a computer involves speaking the language that computer understands, which immediately rules out English as the language of communication with computer.

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. 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 constants, variables and keywords are constructed, and finally how are these combined to form an instruction. A group of instructions would be combined later on to form a program. This is illustrated in the Figure 1.1.



## What is C?

Figure 1.1 English and C Language Similarity

C is a programming language developed at AT& T’s Bell Laboratories of USA in 1972. It was designed and written by a man named Dennis Ritchie. Possibly why C seems so popular is because it is reliable, simple and easy to use

**Integrated Development Environment:** An integrated development environment (IDE), also known as integrated design environment and integrated debugging environment, is a type of computer software that assists computer programmers to write a program.

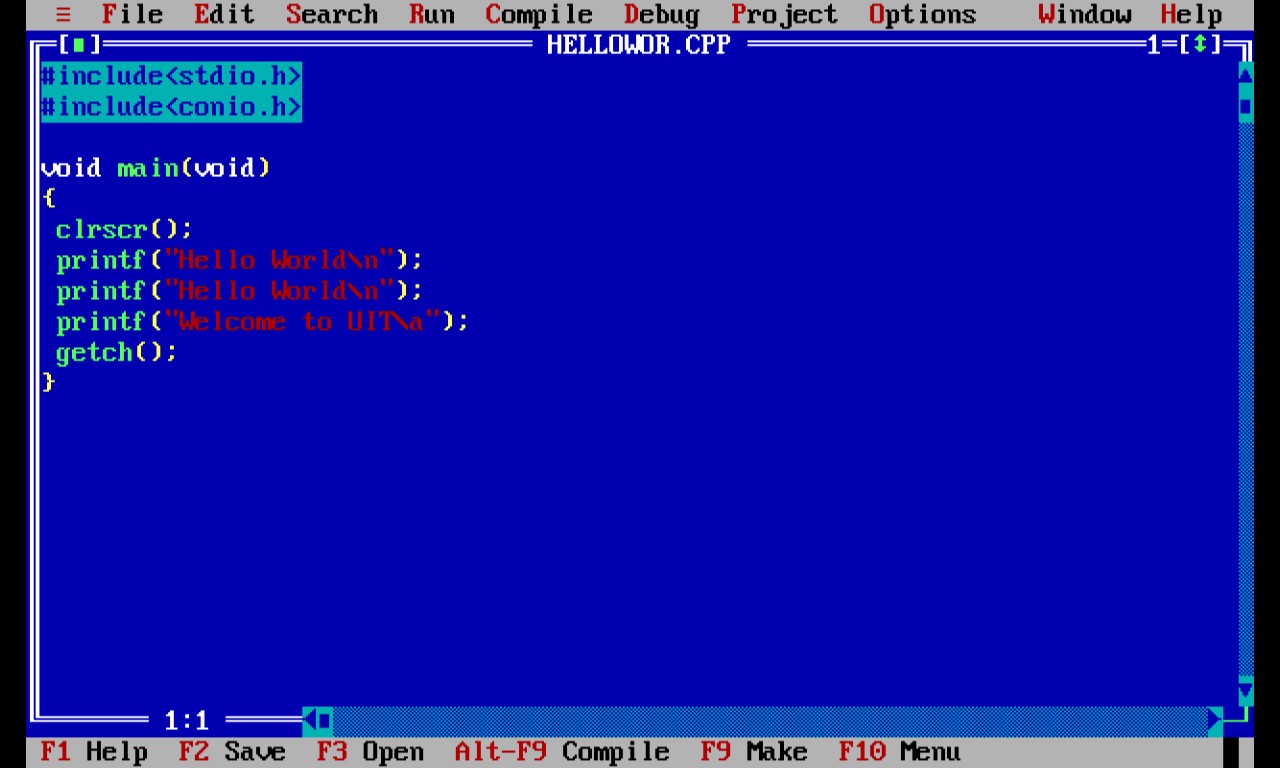


Figure 1.2 Integrated Development Environment

## Some Basic Terminologies

**Instructions:** A line written to perform a specified task on a computer program

**Source Code:** Source Code is any collection of computer instructions (possibly with comments) written using some human-readable computer language, usually as text

**Program**: A computer program, or just a program, is a sequence of instructions, written to perform a specified task on a computer

**Compiler:** A compiler is a computer program (or a set of programs) that transforms source code written in a programming language (the source language) into another computer language (the target language),

**Interpreter:** An Interpreter is a set of instructions written in a [programming](https://en.wikipedia.org/wiki/Programming_language) [language,](https://en.wikipedia.org/wiki/Scripting_language) without previously [compiling](https://en.wikipedia.org/wiki/Compiler) them into a [machine language](https://en.wikipedia.org/wiki/Machine_language) program

**Object File:** An object file is a file containing object code, meaning re-locatable format machine code that is usually not directly executable.

**Linker :**A linker is a [computer program](https://en.wikipedia.org/wiki/Computer_program) that takes one or more [object files](https://en.wikipedia.org/wiki/Object_file) generated by a[compiler](https://en.wikipedia.org/wiki/Compiler) and combines them into a single [executable](https://en.wikipedia.org/wiki/Executable) file, [library](https://en.wikipedia.org/wiki/Library_(computing)) file, or another object file.

**Syntax:** Syntax is the set of rules, principles, and processes that govern the structure of instruction in a given programming language.

## Writing a Program!

In order to write program in any programming language, it is necessary to know what about its **command** and **syntax**. Programmer must also know what is the basic usage of commands and other programming structure. Program written in C language is not very difficult to understand than one written in any other programming language, once you become used to the basic syntax.

Here are the steps that you need to follow to compile and execute your first C program by using Turbo C

1. Double Click Turbo C++ icon from your desktop
2. Select **New** from the **File** menu.
3. Type the program.
4. Save the program using **F2** under a proper name (let say Program1.c).
5. Use **Ctrl + F9** to compile and execute the program.
6. Use **Alt + F5** to view the output.

**Function Definition:** All C programs are divided into units called ‘**functions’**. Every c program consists of one or more functions. Now consider the following program:

#include <stdio.h>

void main(void) { printf( "Hello, World");

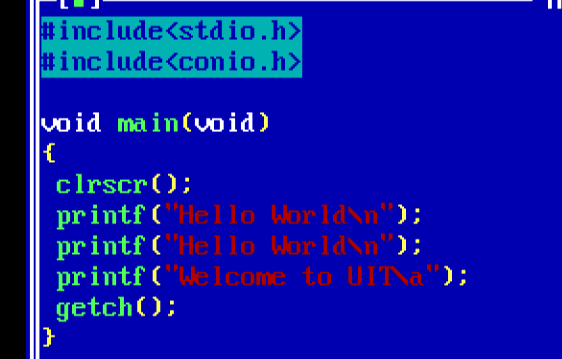
}

Figure 1.3 Simple C Program

It has only one function “**main**”. This function is one to which control is passed from the operating system when the program is run, i.e. it is the first function executed. The word “void” preceding “main” specifies that the function “main” will not return a value. The second “void” in parenthesis specifies that the function takes no arguments.

**Delimiters:** The braces after the function definition signal the beginning and end of the body of the function. The opening brace ({) indicates a block of code that forms a distinct unit is about to begin. The closing brace (}) terminates the block of code.

Braces( {} ) are also used to delimit blocks of code in situations other than function. They are used in loops and decision-making statements with in a program.

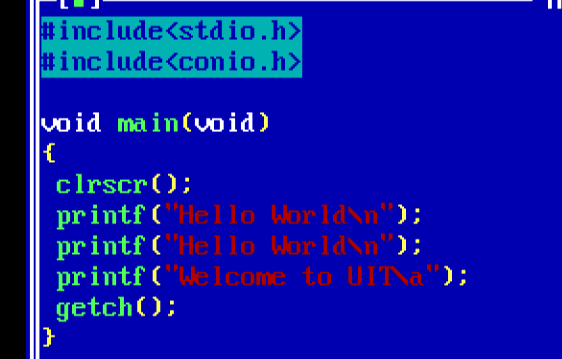


**Delimit**

Figure 1.4 Use of Braces

**Statement Terminator:** A statement in C program in C language is terminated with a **semicolon**, Semicolon terminates the line not the carriage return you type afterwards. C language pays no attention to carriage return in your program listing. The carriage return means the “white spaces” usually created by space or tab.

**Program Indentation:** As you can place as many as white space characters in the program, therefore, they can be used to make the program easier to read. Consider the previous program which can also be written as follows without proper indentation:



**Statement Terminator ;**

Figure 1.5 Use of Terminator

The compiler would not know the difference; however, stretching the code out vertically makes for a more comprehensive program, and aligning and matching braces makes it easier to ensure that each opening brace has a closing brace. Indenting the line of code is not critical in small programs but when there are many sets of nested braces in a program, indentation becomes increasingly important.

## Program Structure

The above prog

Any program may be written according to the following format:

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

void main(void)

{

//Program body

}

Figure 1.6 Format of C program

Figure 1.7 Structure of C Program Explained

Function name

ra

void main(void)

{

printf( "Hello, World");

}

One statement

Closing brace to delimit body of function

Semi-colon terminates each program statement

Opening brace to delimit body of function

The entire program consist of a function called main()

## Output Statement

**printf() :** used to display output of your program The general form of **printf( )** function is,

printf ( “ some text goes here “) ;

**e.g**

printf(“Quaid-e-Azam is our father of nation”);

## Escape Sequence

An **escape sequence** is a sequence of characters that does not represent itself when used inside a character or string literal, but is translated into another character or a sequence of characters that may be difficult or impossible to represent directly

All escape sequences consist of two or more characters, the first of which is the backslash, **\** the remaining characters determine the interpretation of the escape sequence.

|  |  |
| --- | --- |
| **Escape Sequence** | **Meaning** |
| \n | new line |
| \t | tab space |
| \a | alarm |
| \r | carriage return  Returns the cursor to the first character in the same line (ENTER Key). |
| **\\** | to print a Backslash |
| \’ | to print a single quote **‘** |
| \” | double quote |
| \0 | end of string |
| \? | question mark |

## The syntax for comments:

Table 1.1 Escape Sequences

The **//** character is used at the beginning of a single line comment.

The characters **(/\***)is used at the beginning of a multi-line comment, and the character (\*/) is used at the end.

## Pre-lab:

Learn the basic structure for a simple C program

Learn the function of the escape characters and who to use them. Know the syntax for writing comments in the program.

## Assessment Criteria:

The assessment will be based on student’s participation in the lab, if the student solves the exercise, he will get the participation marks for this lab.

## Lab Assignments:

**Lab assignment-1:** Learn how to open Borland Turbo C and open a new C file

**Lab assignment-2-a:** Write the program below to print a simple statement: #include<stdio.h>

#include<conio.h> void main(void)

{

printf ("Welcome to the wonderful world of C/C++!!!\n");

}

**Lab assignment-2-b:** Rewrite the program in assignment 2-a with “getch();” and explain what difference did you detect in the output.

#include<stdio.h> #include<conio.h> void main(void)

{

printf ("Welcome to the wonderful world of C/C++!!!\n"); getch(); //add this, will show the output screen

}

**Lab assignment-2-c:** Insert tab between each word in the printf statement of assignment 2-b, using the appropriate escape sequence.

**Lab assignment-3:** Write a program to print the following text: Hello. (insert tab) Welcome to my first C program.

(empty line)

My name is . (empty line)

I am a student of Dawood University of Engineering & Technology.

**Lab assignment-4:**In assignment 3 add clrscr(); as shown below:

#include<stdio.h> #include<conio.h> void main(void)

{

clrscr();// This will clear the screen

...........

……… //your code written in assignment 3

………

}

**Student Task:**Create a program to print your BIO DATA that include: Your Name

Fathers name Last education Address College Name

Number of Siblings

**Bonus task**: Create a stair case with the help of escape sequences \t and \n, as shown below:

