Unit 1
INTRODUCTION TO PROGRAMMING

Q. What is Computer program? Explain the concept of Computer programming language.

The set of instructions given to the computer are known as a computer program or Software, and the process of feeding or storing these instructions in the computer is known as computer programming. The person who knows how to write a computer program correctly is known as a programmer.

Computer cannot understand English, Urdu or any other common language that humans use for interacting with each other. They have their own special languages, designed by computer programmers. Programmers write computer programs in these special languages called programming languages. Java,

C, C++, C #, Python are some of the most commonly used programming languages.

C language was developed by Dennis Ritchie between 1969 and 1973 at Bell Laboratories

Q. Why do we need a programming environment?

In order to correctly perform any task, we need to have proper tools. For example, for gardening we need gardening tools and for painting we need a collection of paints, brushes and canvas. Similarly, we need proper tools for programming.

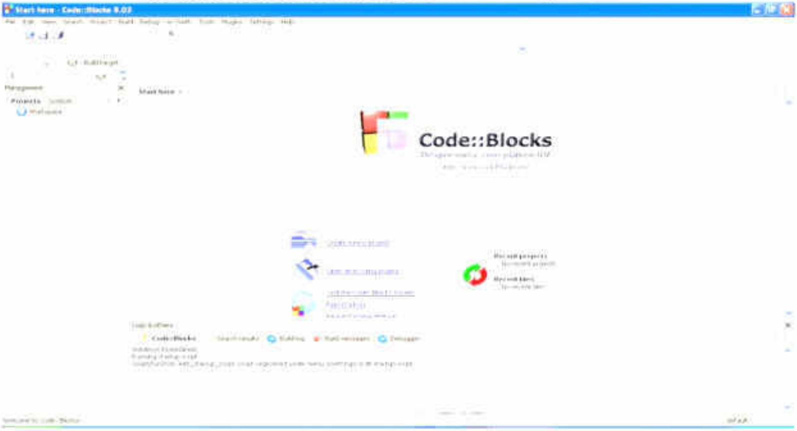
Programmer needs proper tools for programming. A collection of all the necessary tools for programming makes up a Programming environment. It is essential to setup a programming environment before we start writing programs. It works as a basic platform for us to write and execute programs.

Q. What is Integrated Development Environment (IDE)? Write down the IDEs for C programming language.

The Software that provides a programming environment to facilitate programmers in writing and executing computer programs is known as an Integrated Development Environment (IDE). An IDE has a graphical user interface (GUI) meaning that a user can interact with it using windows and buttons to provide input ar it. An IDE consists of tools that help a programmer throughout the phases of writi and testing a computer program. This is achieved by combining text editors, compil ggers in a single interface. Some IDEs for C programming language are:

1) Visual Studio **2) Xcode** **3) Code::Blocks** **4) Dev C++**

The main screen of CodeBlocks IDE is shown in figure below.

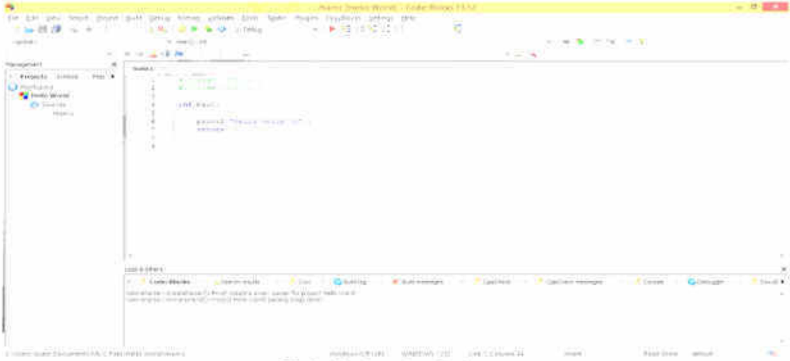


Main interface of code::Blocks

IDE consist of following two parts.

1. **Text Editor**

A text editors is a software that allows programmers to write and edit computer Programs. All IDEs had its own specific text editors. We can write our programs is shown on the main screen of an IDE.

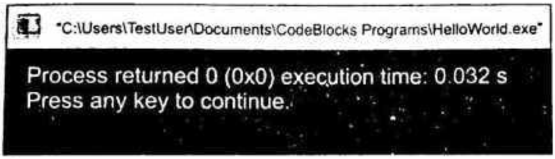


Text editor in code::Blocks

The basic C language program written in the text editor of IDE Code::Blocks. When executed, this program displays Hello World! on computer screen. We have to save our file before it can be executed. We have named our program file as "Hello World" We can click on the build and run button to see the program's output.

Running program in code::Blocks

A console screen showing the output

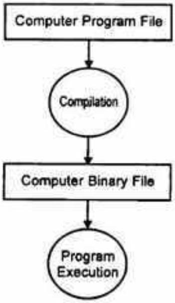


"C:\Users\TestUser\Documents\CodeBlocks Programs\HelloWorld.exe"

```
Process returned 0 (0x0) execution time: 0.032 s
Press any key to continue.
```

2. Compiler

A compiler is software that is responsible for conversion of a computer program written in some high level language to machine code. Computers only understand and work in machine language consisting of 0s and 1s. It convert all code into machine code at same time.



Q. Explain Programming Basics?

(i) Syntax of the language

Each programming language has some primitive building blocks and provides some rules in order to write an accurate program. This set of rules is known as syntax of the language. Syntax can be thought of as grammar of a programming language.

(ii) Syntax Error

While programming, if proper syntax or rules of the programming language are not followed, the program does not compiled. In this case, the compiler generates an error. This kind of errors is called syntax errors.

(iii) Reserved Words

Every programming language has a list of words that are predefined. Each word has its specific meaning already known to the compiler. These words are known as reserved words or keywords. If a programmer gives them a definition of his own, it causes a syntax error. The following reserved words are used in C Programming language.

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Q. Explain the Structure of a C language Program?

A program written in C language can be divided into three main parts:

1. Link section or header section:

While writing programs in C language, we make extensive use of functions that are already defined in the language. But before using the existing functions, we need to include the files where these functions have been defined. These files are called header files. We include these header files in our program by writing the include statements at the top of program.
General structure of an include statement is as follows:

#include < header_file_name >

Here header file name can be the name of any header file in the above example. We have included file stdio.h that contains information related to input and output functions. Many other header files are also available for example file math.h contains all predefined mathematics functions.

2. Main section / Program Body:

It consists of a main function. Every C program must contain a main() function and it is the starting point of execution.

3. Body of main () function:

The body of main() is enclosed in the curly braces { }. All the statements inside these curly braces make the body of main function. In the program the statement printf ("Hell o world!"); uses a predefined function printf to display the statement Hello World! on computer screen.

Rule for writing C language program:

- The sequence of statements in a C language program should be according to the sequence in which we want our program to be executed.
- C language is case sensitive. It means that if a keyword is defined with all small case letters, we cannot capitalize any letter i.e. *int* is different from *Int*.
- Each statement ends with a semi-colon; symbol.

Q. Write down the Purpose and Syntax of Comments in C Programs.

Comments are the statements in a program that are ignored by the compiler and do not get executed. Usually comments are written in natural language e.g. in English language, in order to provide description of our code.

Purpose of writing comments

Comments can be considered as documentation of the program. Their purpose is:

1) It facilitates other programmers to understand our code.

2) It helps us to understand our own code even after years of writing it. We do not want these statements to be executed, because it may cause syntax error as the statements are written in natural language.

Syntax of writing comments

In C programming language, there are two types of comments:

1. Single-line Comments

Single-line comments start with //. Anything after // on the same line, is considered a comment. For example, // This is a comment.

2. Multi-line Comments

Multi line comments start with /* and end at */. Anything between /* and */ is considered a comment, even on multiple lines. For example,

/*this is
a multi- line comment */

Following example code demonstrates the usage of comments:

Example

```
#include < stdio.h >
/*this program displays "I am a student of class 10th" on the output screen*/
void main( )
{ //body of main function starts from here
printf ("I am a student of class 10th");
} //body of main function ends here
```

Q.7. Define Constants and its types.

Each language has a basic set of alphabets (character set) that are combined in an allowable manner to form words, and then these words can be used to form sentences. Similarly, in C programming language we have a character set that includes:

1) Alphabets (A, B, ... Y, Z), (a, b....y,z)

2) Digits (0 - 9)

3) Special symbols (~ ' ! @ # % ^ & * ( ) _ - + = ! \ { } [ ] :; " < >,. ? /)

Constants

Constants are the values that cannot be changed during the execution of a program e.g. 5, 75.7, 1500 etc. In C language, there are three types of constants:

1. Integer Constants **2. Real Constants** **3. Character Constants**

Integer Constants:

These are the values without a decimal point e.g. 7, 1256, 30100, 53555, -54, -2349 etc. It can be positive or negative. If the value is not preceded by a sign, it is considered as positive.

Real Constants:

These are the values including a decimal point e.g. 3 .14, 15.3333, 75.0, -1575.76, -7941.2345 etc. It can also be positive or negative.

Character Constants:

Any single small case letter, upper case letter, digit, punctuation mark, special symbol enclosed within ' ' is considered a character constant e.g.'5', '7' , 'a', 'X', '.' etc.

Difference between Integer and Character Constant:

A digit used as a character constant i.e. '9', is different from a digit used as an integer constant i.e. 9. We can add two integer constants to get the obvious mathematical result e.g 9 + 8 = 17, but we cannot add a character constant to another character constant to get the obvious mathematical result e g '9' + '8' ≠ 17.

Q. What is Variables? Explain Data Type of a Variable.

Variables:

A variable is actually a name given to a memory location, as the data is physically stored inside the computer's memory. The value of a variable can be changed in a program. It means that, in a program, if a variable contains value 5, then later we can give it another value that replaces the value 5. Some examples of valid variable names are height, Average Weight, _var1.

Each variable has a unique name called identifier and has a data type. Data type describes the type of data that can be stored in the variable. C language has different data types such as int, float, and char. Following table shows the matching data types in C language,against different types of data.

Type of data	Matching data type in C language	Sample value
Integer	int	123
Real	float	23.5
Character	char	'a'

Data Type of a Variable

Each variable in C language has a data type. The data type not only describes the type of data to be stored inside the variable but also the number of bytes that the compiler needs to reserve for data storage. The data types used in C language are:

Integer (int)

Integer data type is used to store integer values (whole numbers). Integer takes up 4 bytes of memory. To declare a variable of type integer, we use the keyword int. there are two types of integers.

1. **Signed int**

A signed int can store both positive and negative values ranging from 2,147,483,648 to 2,147,483,647. By default, type int is considered as a signed integer.

2. **Unsigned int:**

An unsigned int can store only positive values and its value ranges from 0 to +4,294,967,295. Keyword unsigned int is used to declare an unsigned integer.

Floating Point (float)

Float data type is used to store a real number (number with floating point) up to six digits of precision. To declare a variable of type float, we use the keyword float. A float uses 4 bytes of memory. Its value ranges from 3.4×10^{38} to 3.4×10^{-38}.

Character (char)

To declare character type variables in C, we use the keyword char. It takes up just 1 byte of memory for storage. A variable of char type can store one character only.

Q. Write down the rules for writing the name of variables in C language?

Each variable must have a unique name or identifier. Following rules are used to name a variable.

1. A variable name can only contain alphabets (uppercase or lowercase) digits and underscore sign.
2. Variable name must begin with a letter or an underscore, it cannot begin with a digit.
3. A reserved word cannot be used as a variable name.
4. There is no strict rule on how long a variable name should be, but we should choose a concise length for variable name to follow good design practice.
5. We should give appropriate name to a variable, that describes its purpose e.g. in order to store salary of a person, appropriate variable name could be salary or wages.

Q. What is difference between Variables and Constants?

Variables:

A variable is actually a name given to a memory location, as the data is physically stored inside the computer's memory. The value of a variable can be change during the execution of a program. It means that, in a program, if a variable contains value 5, then later we can give it replaces the value 5. Some examples of valid variable names are height, Average, Weight, _var1.

Constants:

Constants are the values that cannot be changed during the execution of a program e.g. 5, 75.7, 1500 etc. In C language, there are three types of constants:

1. Integer Constants **2. Real Constants** **3. Character Constants**

Q. How can we declare and Initialize a variable?

Variable Declaration:

We need to declare a variable before we can use it in the program. Declaring variable includes specifying its data type and giving it a valid name. Following syntax can be followed to declare a variable.

Data_ type variable_ name;

Some examples of valid variable declarations are as follows:

unsigned int age;
float height;
int salary;
char marital_status;

Multiple variables of same data type may also be declared in a single statement as shown in the following examples:

unsigned int age, basic_salary, gross_salary;
int points_scored, steps;
float height, marks;
char marital_status, gender;

A variable cannot be declared unless we mention its data type. After declaring a variable, its data type cannot be changed. Declaring a variable specifies the type of variable, the range of values allowed by that variable, and the kind of operations that can be performed on it. Following example shows a program declaring two variables:

Example

```
void main ()
{
char grade;
int value;
}
```

Variable Initialization

Assigning value to a variable for the first time is called variable initialization. C language allows us to initialize a variable both at the time of declaration, and after declaring it. For initializing a variable at the time of declaration, we use the following general structure.

data_type variable_name = value ;

Following example shows a program that demonstrates the declaration and initialization of two variables.

Example

```
#include<stdio.h>
void main()
{
char grade; //Variable grade is declared
int value = 25; /*Variable value is declared and initialized. */
grade = 'A'; //Variable grade is initialized
}
```