


# Programming Fundamentals

## Unit 1

### INTRODUCTION

In order to communicate with each other, we use natural languages like Bengali, English, Hindi, Urdu, French, Gujarati etc. We have different language around us in our real life to communicate with each other. Basically, human beings cannot speak or write in computer language, and since computers cannot speak or write in human language, an intermediate language had to be devised to allow people to communicate with the computers. These intermediate languages, known as programming languages, allow computer programmer to direct the activities of the computer. In this unit, we shall be exposed to general terms related to computer programming such as languages, algorithms and flowcharts. You will be familiar with these terms after reading this unit. The importance of an algorithm in a programming language can be realized. This unit overviews history of C language in addition basic structure of a C program is discussed.


<p>Timeframe</p>  <p>How long ?</p>	<p>We expect that this unit will take maximum 8 hours to complete.</p>
--	--

<p><b>Unit Structure</b></p>	
	<p>Lesson- 1 : Introduction to Computer Programming</p> <p>Lesson- 2 : Algorithm and Flowchart</p> <p>Lesson- 3 : Overview of C language</p> <p>Lesson- 4 : Basic structure of C</p>

**Lesson-1****Introduction to Computer Programming****Learning Outcomes****Outcomes**

Upon completion of this lesson you will be able to

- Define computer programming.
- State Why Programming.
- State the Categories of programming language.
- Define machine, assembly and high-level language.

	<b>Keywords</b>	<b>Computer, Language, Program, Assembly, Machine</b>
---	-----------------	---

Hope all of you are conscious with the term language. Language is a system of communication between two persons. Observe, In order to communicate with each other, we use languages like Bengali, English, Hindi, Urdu, French etc. We have different language around us in our real life to communicate with each other. However, human beings cannot speak or write in computer language, and since computers cannot speak or write in human language, so an intermediate language had to be devised (design) to allow people to communicate with the computers. These intermediate languages, known as Computer Language, that allow a computer programmer to direct the activities of the computer.

**COMPUTER PROGRAM**

Let us first recall computer and what they do.

- A computer is an electronic device or machine used for processing data to deliver required information.
- A computer is a programmable electronic machine that receives input, stores and process data, and provides output in a useful format.
- Computer has no intelligence of its own to perform any task.
- We also know that hardware (Computer) is all of the physical equipments or components which are visible and touchable. A computer without a computer program (software) is just a dump box; it is *programs* that make computers active. The usefulness of the computer depends on the programs that are written to manipulate it.

As you know computer cannot do anything on its own and has to be guided (directed) by the user. In order to do any specific job you have to give a sequence of instructions to the computer. This set of instructions is called a computer program. Computer program is a sequence of instructions written in a Computer Language to perform a specified task by the computer. Computer Programming is the process of developing and implementing various sets of instructions to enable a computer to do a certain task.



Note it!

---

*Program is a series of instructions that guides the computer to solve a particular problem.*

---

**WHY PROGRAMMING?**

Hope all of us know that computer can neither think nor make any judgments on its own; it can't do anything by themselves. They do exactly what you tell them to do: no more, no less unlike human beings. Suppose, if someone asks you, "What is the sum of 2+2?" or "add 2+2=?" or just, "2+2=?" you understand anyway but computer is different. Instructions to the computer should be explicitly stated. So we can say that computer needs a program to tell it what to do.



Activity

1. Mention the significance of Computer programming by your own concept.  
.....  
.....  
.....

**PROGRAMMING LANGUAGE**

In earlier section we have learnt about computer language, computer program. In this section we will learn about programming language. We have different language around us in our real life to communicate with each other. What do you think if a Bengali person talks to an English man in Bengali language who does not know Bengali, can they communicate with each other? I am sure all of you will say no. In the same way if you tell the computer do this work or koro ei kaj ta, computer will not understand anything because computer understand only its own language (0 and 1). That's why need programming language. To build programs, people use languages which are similar to human languages that are translated into machine code, which computers understand. A Programming Language (Computer) is composed of a set of characters, words and rules of the language that can be used to perform specific tasks by writing a computer program. Examples of some of the programming languages are BASIC, FORT RAN, COBOL, C, C++ etc.

There are three types of programming language according to their levels and they are as follows:

- { Machine Language } Low Level Language – Machine Oriented
- { Assembly language }
- High Level Language

**MACHINE LANGUAGE**

- Machine language is the only language that is directly understood by the computer. It is written in binary form that is 0 and 1.

It is most efficient for the computer as the instructions are directly executed.

- It is not an easy language to learn because of its complexity as it consists of 1's and 0's. Program instruction may look like this: 1011000111101
- Machine language is hardware dependent. A program written for one computer might not run on other computers with different hardware configuration. It is also difficult to debug the program written in this language.
- The only advantage is that program of machine language run very fast because no translation program is required for the CPU.
- It is considered to the first generation language.

The following program is an example of a machine language program for adding two numbers.

```

0011 1110 Load A register with value 7
0000 0111 value 7
0000 0110 Load B register with 10
0000 1010 A= A + B
1000 0000 Store the result
0011 1010 into the memory location
0110 0110
0000 0000 whose address is 100 (decimal)
0111 0110 Halt processing

```

### ASSEMBLY LANGUAGE

- In assembly language, the operation code is expressed as a combination of letters rather than binary numbers, sometimes called mnemonics. This allows the programmer to remember the operations codes easily than when expressed strictly as binary numbers.
- Translator program (called Assembler) translate the programs written in assembly language into machine language for execution by the computer.
- It is relatively easier to correct errors and modify program instructions.
- Assembly Language has almost the same efficiency of execution as the machine language.
- Assembly language is machine dependent. A program written for one computer might not run on other computers with different hardware configuration.
- It is considered to be a second generation language.

Example of an assembly language program for adding two numbers as follows:

```

LDA, 7 Load register A with 7
LDB ,10 Load register B with 10
ADD A,B AA + B
LD(100), A Save the result in the location 100
HALT Halt process

```

### HIGH LEVEL LANGUAGE

- High Level Languages are user-friendly languages which are similar to English with vocabulary of words and symbols.
- High level languages are simple language that use English and mathematical symbols like +, -, %, /, etc. for its program construction.
- High level language has to be converted to machine language for the computer to understand.
- High level languages are easy to learn and use. It is because that they are similar to the languages used by us in our day-to-day life.
- The code is then translated into object code, using a translator called Compiler or Interpreter. The programs can easily be debugged and are machine independent.
- For Example: FORTRAN, COBOL, PASCAL C, C++, Java, C#, Visual Basic etc.

**Lesson-2****Algorithm and Flowchart****Learning Outcomes****Outcomes**

Upon completion of this lesson you will be able to

- Define algorithm.
- Write algorithm.
- Define flowchart.
- Explain Flowchart symbols & their meanings.
- Draw flowchart.

	<b>Keywords</b>	<b>Algorithm, flowchart, Symbol</b>
---	-----------------	-------------------------------------

**ALGORITHM**

From programming point of view,

- An algorithm is a step by step procedure (technique) for solving any problem.
- An algorithm is an effective method expressed as a finite set of well defined instructions.
- Refers to the logic of a program and step by step description of how to arrive at the solution of a given problem.

*For example, find the sum of two numbers.*

```

Step 1: Start
Step 2: Input A
Step 3: Input B
Step 4: Calculate C = A + B
Step 5: Output C
Step 6: Stop

```

Another example, *Consider the following instructions that you need to boiling water.*

```

Step 1: Start
Step 2: Take water in a kettle.
Step 3: Place the kettle on the oven.
Step 4: Turn the oven on.
Step 5: Check whether water is boiled
Step 6: Turn the oven off
Step 7: Stop


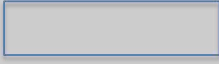

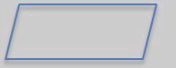


```

**FLOWCHART**

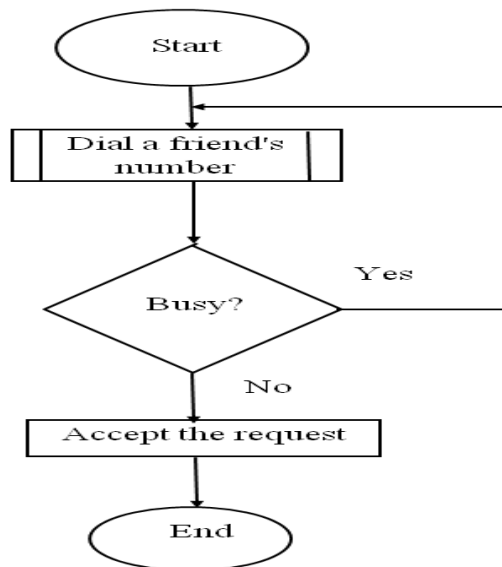
- A flowchart is a graphical/ pictorial representation of an algorithm. a flowchart is most helpful in planning, designing, and structuring a program.
- Uses symbols (boxes of different shapes) that have standardized meanings to denote different types of instructions.

- Process of drawing a flowchart for an algorithm is called flowcharting.
- Boxes are connected by solid lines having arrow marks to indicate exact sequence in which the instructions are to be executed.

**Symbols of Flowchart**

Start and end blocks. looks like	
Process or Operation block which gives instructions for work to be done. looks like	
Decision block (logical block) which asks a question and has two or more outputs. looks like	
Input / output are represented by specially shaped blocks	
Some times the same job may be repeated in different places in flowchart. modularity in flow chart. looks like	
Flow lines	

For example



**Advantages of Flowchart**

- Efficient coding
- Systematic debugging
- Systematic testing
- Proper documentation

**Lesson-3****Overview of C language****Learning Outcomes****Outcomes**

Upon completion of this lesson you will be able to

- Write history of C.
- Explain the importance of C.
- Write application of C.
- Define compiler and interpreter.

	<b>Keywords</b>	<b>History, Unix, Portable</b>
--	-----------------	--------------------------------

**BRIEF HISTORY OF C**

- C is a fast, general-purpose, platform independent, high-level programming language that evolved by “Dennis Ritchie” at Bell Laboratories from two previous programming languages, BCPL and B on DEC PDP-11 machine in 1972. Initially, C widely known as the development language of the UNIX operating system but today all new major operating systems are written in C and/or C++.

Evolved over the years

- 1972 – C invented
- 1978 – *The C Programming Language* published; first specification of language
- 1989 – C89 standard (known as ANSI C or Standard C)
- 1990 – ANSI C adopted by ISO, known as C90 and 1999 – C99 standard
- 2007 – work on new C standard C1X announced

**WHY C**

- C is a powerful and flexible language.
- C is the most widely used and popular System Programming Language.
- C has rich set of built-in functions and support variety of data types & operators.
- C is highly portable (Machine independent) and it is static (compiled), typed, structured and imperative.
- C has ability to extend itself and it is stable (the language doesn't change much).
- C is quick running (code written in c is efficient & fast).
- C is the basis for many other languages (C++, C#, Java, Perl etc).
- C is small (only 32 keywords) and it is a Programmers Language.

**APPLICATIONS OF C**

- Systems Programming: *e.g.*, compilers and interpreters
- Operating System: *e.g.*, Linux ,Unix
- Microcontrollers: automobiles and airplanes
- Embedded Processors: phones, portable electronics etc.
- Database systems and DSP Processors: digital audio and TV systems.

**CREATION, COMPI LATION AND EXECUTION OF C PROGRAM****Environment Setup**

Though Environment Setup is not the part of Programming Language, but it is the first step to be followed before setting on to write a program. Thus, we need to have the required software setup, i.e., installation on our PC which will be used to write computer programs, compile, and execute them. Once C compiler is installed in your machine, you can create, compile and execute C programs. You need the following two software tools available on your computer to set up your environment for starting C programming language,

- (i) Text Editor / Code editor and
- (ii) C Compiler.

You are able to write program in Notepad but, notepad and the other editors don't provide you with the right tools to do software development. That's why code editor or an IDE is better option instead of notepad. Code editor is a piece of software that you can use to write your code and it will behave like the tool you need to develop software, for example, coloring keywords, display the line numbers that's important for debugging, allow for multiple text manipulation and add a file explorer or project explorer to select the files you need to work on. Beside, a compiler is necessary to compile the program which converts your program (source code) into a language understandable by a computer. So before proceeding, make sure you have C compiler available at your computer. There are many compilers available on the internet and sold commercially in stores or online. Most compilers come with a built-in code editor that can be used to enter source code; however, some don't.



Note it!

---

***When you save a source file, you must give it a name. The name should describe what the program does. In addition, when you save C program source files, give the file a .C extension.***

---

## COMPILER AND INTERPRETER

As we write the code in English and we know that computers can understand only 0s and 1s. So we need a translator which translates the code of our program into machine language. There are two kinds of translators which are known as Interpreter and Compilers. These translators translate our program which is written in C-Language into Machine language.

Compilers also translate the English like language (Code written in C) into a language (Machine language) which computers can understand. The Compiler read the entire (whole) program first and then translates it into machine language completely. The program written by the programmer in higher-level language is called source program. After this program is converted to machine language by the compiler it is called object program. A compiler can translate only those source programs, which have been written, in that language for which the compiler is meant for. For example, FORTRAN compiler will not compile source code written in COBOL language. Interpreters translates the program line by line that means it reads one line of program and translates it into machine language and executes it immediately, then it reads second line, translate it into machine language and executes it immediately and so on. Here translation and execution are carried out for each statement. It differs from compiler, which translate the entire source program into machine code and then involve in its execution.

The benefit of it is that we get the errors as we go along and it is very easy to correct the errors. The drawback of the interpreter is that the program executes slowly as the interpreter translates the program line by line. Another drawback is that as interpreters are reading the program line by line so they cannot get the overall picture of the program hence cannot optimize the program making it efficient.

The difference between interpreter and compiler is that compiler will stop translating if it finds an error and there will be no executable code generated whereas Interpreter will execute all the lines before error and will stop at the line which contains the error. So Compiler needs syntactically correct program to produce an executable code.



**Lesson-4****Basic Structure of C Program****Learning Outcomes****Outcomes**

Upon completion of this lesson you will be able to

- Explain basic element of C program.

**Keywords****Comments, printf, scanf, Compound, Statement**

Let us look at a simple program (code) that will print a Line of Text

```

/* first program in C */
#include <stdio.h>
int main()
{
    printf( "Welcome to C" );
    return 0;
}

```

**Output**

Welcome to C

Let's look at the elements of the program.

```
/* first program in C */
```

Known as comments. Comments are like helping text in your C program and they are ignored by the compiler and Can appear almost anywhere. They start with /\* and terminate with the characters \*/

```
#include <stdio.h>
```

#include, the first word of any C program. It is also known as pre-processor. The main work of pre-processor is to initialize the environment of program, i.e to link the program with the header file <stdio.h>. Standard header files are provided with each compiler, and cover a range of areas like string handling, mathematical functions, data conversion, printing and reading of variables. To use any of the standard functions, the appropriate header file must be included. This is done at the beginning of the C source file. For example, to use the printf() function in a program, the line #include <stdio.h> is responsible.

**main()**

The next important line is main(). main() function is a function that must be used in every C program. main() function starts the execution of C program. Parentheses “( )” is used to indicate a function.

Braces { and } indicate a block. The bodies of all functions must be contained in braces.

**printf( "Welcome to C" );**

The printf() is a library function to send formatted output to display on the screen. Printf() instructs computer to perform an action; specifically, prints the string of characters within quotes (“ ”). There is a semicolon (;) at the end of the above statement. In a C program, the semicolon is a statement

terminator; each individual statement must be ended with a semicolon (;). This is very important missing of a semicolon (;) at the end of statement is a syntax error and compiler will report an error during compilation.

### **return 0;**

A way to exit a function., in this case terminates the main() function and returns the value 0




---

***Don't forget to have a main function, or mistype the name, the compiler will give you an error.***

---

## **INPUT/ OUTPUT STATEMENT**

In this section we will learn how a computer reads data (input) and prints data (output). Two commonly used functions for I/O (Input/Output) are scanf() and printf().

### **scanf() function**

scanf() function reads one or more values from an input device, and stores them into variables specified by the programmer. In scanf() syntax a special character format specifier is used to specify the data type of value.

- In C programming language, scanf() function is used to read character, string, numeric data from keyboard.
- The format specifier %d is used in scanf() statement. So that, the value entered is received as an integer and %s for string.
- Ampersand (&) is used before variable name in scanf() statement as &number.

### **Syntax**

```
scanf("format specifiers",&value1,&value2,.....);
```

### **Example**

```
int a;
float b;
char c;
scanf("%d%f %c",&a,&b,&c);
```

In the above example scanf() is able to read three input values ( integer, float value and character) and those are stored in a, b and c variable respectively.

Format specifier:

<b>Format specifier</b>	<b>Type of value</b>
%d	Integer
%f	Float
%lf	Double
%c	Single character
%s	String
%u	Unsigned int
%ld	Long int
%lf	Long double

### **printf() function**

Printf is a predefined function in "stdio.h" header file, by using this function, we can display the data or user defined message on monitor.

- In C programming language, printf() function is used to display the “character, string, float, integer, octal and hexadecimal values” onto the output screen. While working with printf(), it can take any number of arguments but first argument must be within the double cotes (" ") and every argument should separated with comma ( , ) Within the double cotes, whatever we pass, it prints same.

- We use printf() function with %d format specifier to display the value of an integer variable. Similarly %c is used to display character, %f for float variable, %s for string variable, %lf for double and %x for hexadecimal variable.
- To create a newline, we use “\n” in C printf() statement.

Syntax

```
printf("user defined message");
printf("Format specifiers",value1,value2,..);
```

Example of printf() function

```
int a=10;
double d=13.4;
char c='A';
printf("%f%d",d,a,c);
```


**SIMPLE AND COMPOUND STATEMENT**

Statements can be either simple or compound. A simple statement performs a single action. Examples of simple statements are as follows:

```
printf( "Welcome to C" );
int i = 12;
```

Compound statement does several actions in sequence. A compound statement is composed of simple statements enclosed within a pair of braces { }. The following is an example of compound statements in C.

```
{
pi= 3.141593;
area= pi*radius*radius;
}
```

 <b>Summary</b>	
<b>In this lesson</b>	
<ul style="list-style-type: none"> <li>▪ We have understood C programming language and programming language types.</li> <li>▪ Also understood difference between compiler and interpreter.</li> <li>▪ Also understood various C syntax, flowchart and formate specefier.</li> </ul>	

**Assessment**



Assessment

**Fill in the blanks**

1. .... developed the C language.
2. printf() is used to ..... the output and scanf() is used to ..... the input.
3. C programs are stored with ..... Extension.

**Multiple Choice Questions (MCQ)**

1. What is a program?
  - a) set of instruction      c) set of algorithm
  - b) set of pseudo code    d) none of these
2. Which of the following is a graphical representation of a program’s flow?

- a) algorithm
- b) flowchart
- 3. The C language has been developed at?
  - a) AT & T Bell Labs
  - b) Borland International
- c) control of flow
- d) UML diagram
- 4. What is the only function that all C programs must contain?
  - a) start()
  - b) main()
- 5. Which of the following is the correct way of writing comments?
  - a) `*/comments/*`
  - b) `/*comment*/`
  - c) `**comment**`
  - d) `{comment}`

**Exercises**

1. Define computer programming language.
2. Define machine, assembly and high-level languages.
3. Differentiate between an interpreter and compiler.
4. Compare an algorithm with a flowchart.
5. Explain the symbols for flowchart.
6. Write the syntax of printf and scanf function.
7. Define simple and compound statement.
8. Write the format specifier available in C language.