

# 3

## SOFTWARE AND PROGRAMMING LANGUAGES

### 3.1 INTRODUCTION

In the previous lesson we discussed about the different parts and configurations of computer. It has been mentioned that programs or instructions have to be fed into the computer to perform specific task. It is therefore necessary to provide instructions to the computer in a systematic order to complete our task. We can divide the computer components into two major areas, namely, hardware and software. Hardware is the machine itself and its various individual equipment. It includes all mechanical, electronic and magnetic devices such as monitor, printer, electronic circuit, floppy and hard disk. In this lesson we will discuss about the other part, namely, software that makes use of hardware for performing various functions.

### 3.2 OBJECTIVES

After going through this lesson you would be able to:

- explain the concept of software
  - distinguish between different types of software
  - differentiate application software from system software
  - differentiate between different types of language
  - distinguish between compiler and interpreter
-

### 3.3 WHAT IS SOFTWARE?

As you know computer cannot do anything on its own and has to be guided 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. Software refers to the set of computer programs, procedures that describe the programs and how they are to be used. We can say that it is the collection of programs, which increases the capabilities of the hardware. Software guides the computer at every step where to start and stop during a particular job. The process of software development is called programming.

You should keep in mind that software and hardware are complementary to each other. Both have to work together to produce meaningful result. Another important point you should know that developing software is difficult, time consuming and expensive.

### 3.4 SOFTWARE TYPES

Computer software is normally classified into two broad categories.

- Application Software
- System Software

**3.4.1 Application Software:** Application software is a set of programs, which are written to perform specific tasks of the users of computers such as Accounts, Stores, Payroll, etc. These software are developed in high level language to help the user to get the computer perform the tasks. For example, you can develop a package to print mark sheet of every student of your class or generate accounts of a company, etc. Application software can be classified into two types:

(a) Customized Packages      (b) Generalized Packages.

**Customized Packages:** These are developed especially for the user by a program using high-level computer languages. For example, when you develop a package to prepare a status report of your school i.e., name of the students, their addresses, Parent's name, Fee paid, marks obtained, etc., you are developing a customized package because the package developed for your school may not be of any use for other school.

**Generalized Package:** These packages are written for the people who have to perform common task on a computer system. DBase, Lotus 1-2-3, FoxPro, MS Office, etc. are the names of the Generalized Packages. Any person can use these packages because they

---

can be used for different application and purposes. These are meant for mass consumption.

Another example of application software is programming language. Among the programming languages COBOL (Common Business Oriented Language) is more suitable for business application whereas FORTRAN (Formula Translation) is useful for scientific application. We will discuss about these languages in next section.

**3.4.2 System Software:** You know that an instruction is a set of programs that has to be fed to the computer for operation of computer system as a whole. When you switch on the computer the programs stored or written in ROM is executed which activates different units of your computer and makes it ready for you to work on it. This set of program can be called System Software. Therefore system software may be defined as a set of one or more programs designed to control the operations of computer system.

System Software are general purpose programs designed for performing tasks such as controlling all operations required to move data into and out of the computer. It communicates with keyboard, printer, card reader, disk, tapes etc. It also monitors the use of various hardware like memory, CPU etc. System software acts as an interface between hardware and application software. System software allows application packages to be run on the computer with less time and effort. *Remember that it is not possible to run application software without system software.*

Development of system software is a complex task and it requires extensive knowledge of computer technology. Due to its complexity, the users cannot develop it. Computer manufactures build and supply this system software with the computer. DOS, UNIX, WINDOWS, Language Compilers and Interpreters are some of the widely used system software. Out of these UNIX is a multi-user operating system whereas DOS and WINDOWS are PC-based. We will discuss in detail about DOS and WINDOWS in the next module.

So without system software it is impossible to operate your computer. The fig.3.1 shows relation between hardware, software and you as a user of computer system.

---

### **INTEXT QUESTIONS**

---

1. What are program, programming and software?
2. Differentiate between System Software and Application Software.
3. Write True or False.
  - (a) The set of instructions given to the computer is called programming.
  - (b) Application Software is a set of programs to carry out operations for a specific application.
  - (c) UNIX is a multi-user operating system.

---

### **3.5 PROGRAMMING LANGUAGE**

You are aware with the term language. It is a system of communication between two persons. Some of the basic natural languages that we are familiar with are English, Hindi, Oriya, etc. These are the languages used to communicate among various categories of persons. But how will you communicate with your computer. Your computer will not understand any of these natural languages for transfer of data and instruction. So there are computer-programming languages specially developed so that you could pass your data and instructions to the computer to do specific job. You must have heard names like FORTRAN, BASIC, COBOL, etc. These are programming languages. So instructions for performing a task are written in a particular computer programming language based on the type of job. As an example, for scientific application FORTRAN and C languages are used. On the other hand, COBOL is used mainly for business application.

---

### 3.5.1 Types of Programming Languages

There are two major types of programming languages. These are Low Level Languages and High Level Languages. Low Level languages are further divided into Machine language and Assembly language.

### 3.5.2 Low level languages

The term low level means closeness to the way in which the machine has been built. Low Level languages are machine oriented and require extensive knowledge of computer hardware and its configuration. The low level languages are:

**Machine Language:** Machine Language is the language of the computer and is the only language that is directly understood by the computer. We also call it machine code and it is written as strings of 1's and 0's. It is on this basis that the computer is designed. When this sequence of codes is fed to the computer, it recognizes the codes and converts it in to electrical signals needed to run it. For example, a program instruction may look like this:

1011000111101

It is not an easy language for you to learn because of its complexity as it consists of 1's and 0's. It is most efficient for the computer as the instructions are directly executed. It is considered to the first generation language. It is also difficult to debug the program written in this language.

#### **Advantage**

The only advantage is that program of machine language run very fast because no translation program is required for the CPU.

#### **Disadvantages**

1. It is very difficult to program in machine language. The programmer has to know details of hardware to write program.
2. Machine language is hardware dependent.
3. The programmer has to remember a lot of codes to write a program, which results in program errors.
4. It is difficult to debug the program.

**Assembly Language:** It is the first step to improve the programming

---

structure. In this language, the machine codes comprising of 1's and 0's are substituted by symbolic codes (called mnemonic codes) to improve their understanding.

The set of symbols and letters forms the assembly language and a translator program (called Assembler) is required to translate the programs written in assembly language into machine language for execution by the computer. It is considered to be a second-generation language.

**Advantages:**

1. The symbolic programming of Assembly Language is easier to understand and saves a lot of time and effort of the programmer.
2. It is relatively easier to correct errors and modify program instructions.
3. Assembly Language has almost the same efficiency of execution as the machine level language because this is one-to-one translator between assembly language program and its corresponding machine language program.

**Disadvantages:**

One of the major disadvantage is that assembly language is machine dependent. A program written for one computer might not run on other computers with different hardware configuration.

---

**INTEXT QUESTIONS**

---

4. What is the difference between FORTRAN and COBOL?
5. Differentiate between Machine Language and Assembly Language.
6. Write True or False.
  - (a) Low Level language and High-level language are two major types of programming languages.
  - (b) Machine language is the only language that is indirectly understood by the computer.
  - (c) Assembly language is second-generation language.

---

**3.5.3 High Level Languages**

You know that assembly language and machine language require

---

extensive knowledge of computer hardware. To overcome this limitation, a user writes the instructions in English like sentences to perform the logic of the problem irrespective of the type of computer you are using. The language used for this is referred to as high-level language.

High-level languages are simple language that use English and mathematical symbols like +, -, %, /, etc. for its program construction.

You should know that any higher-level language has to be converted to machine language for the computer to understand.

Higher-level languages are problem-oriented languages because the instructions are suitable for solving a particular problem. For example COBOL (Common Business Oriented Language) is mostly suitable for business oriented language where there is very little processing and huge output. There are mathematical oriented languages like FORTRAN (Formula Translation) and BASIC (Beginners All-purpose Symbolic Instructions Code) where very large processing is required.

Thus a problem-oriented language is designed in such a way that its instruction may be written more like the language of the problem. For example, businessmen use business term and scientists use scientific terms in their respective languages.

### **Advantages of High Level Languages**

Higher-level languages have a major advantage over machine and assembly languages that higher-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 programs can easily be debugged and are machine independent.

### **3.6 COMPILER**

It is a program that translates the instructions of higher-level languages to machine language. It is called compiler because it compiles every program instruction given in higher-level languages into machine language. Thus compiler is a program translator like assembler but more sophisticated. It scans the entire program first and then translates it into machine code.

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.

---

***Fig. 3.2: Computer 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.

Object program generated by compiler is machine dependent. It means programs compiled for one type of machine will not run in another type. Therefore every type of machine must have its personal compiler for a particular language. Machine independence is achieved by using standard higher-level language on different machines and converting them for use on specific machines through a compiler.

### **3.7 INTERPRETER**

An interpreter is another type of program translator used for translating higher-level language instructions into machine language instructions. It takes one statement of higher-level language at a time, translates it into machine language and executes it immediately. 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 advantage of interpreter compared to compiler is its fast response to changes in source program. It eliminates the need for a separate compilation after changes to each program. Interpreters are easy to write and do not require large memory in computer. The

---

disadvantage of interpreter is that it is time-consuming method because each time a statement in a program is executed, it is first translated. Thus compiled machine language program runs much faster than an interpreted program.

---

### **INTEXT QUESTIONS**

---

7. What is the difference between interpreter and compiler?
  8. Give some examples of high-level language.
  9. Write True and False for the following statements
    - (a) High level languages are problem-oriented language.
    - (b) Object program generated by compiler is machine independent.
    - (c) The disadvantage of interpreter is that it is time consuming.
- 

### **3.8 WHAT YOU HAVE LEARNT**

In this lesson we discussed about two types of software, namely, system software and application software. System software controls the hardware part of the computer. It is designed for performing tasks such as controlling all operations required to move data into and out of the computer. It communicates with printer, card reader, disk, tapes, etc. and monitors the use of various components like memory, CPU, etc. Application software is a set of programs written for specific purpose. Examples of application software are MS WORD, Lotus 1-2-3, Tally Accounting software, etc. We also discussed about levels of computer language.

---

### **3.9 TERMINAL QUESTIONS**

---

1. Describe the concept of software and hardware.
  2. Define computer Language.
  3. State the three different categories of computer languages.
  4. What is machine language? Why is it required?
  5. Describe the advantages and disadvantages of machine languages.
  6. What is assembly language? Explain its advantages over machine languages.
-

7. Differentiate between
  - (a) Source program and object program
  - (b) Higher level language and machine language
  - (c) Compiler and Interpreter

---

### **3.10 KEY TO INTEXT QUESTIONS**

---

1. Program is a set of instructions given to the computer by the user. Software is a set of computer programs and procedure that describe the programs. Programming is the process of software development.
  2. Application Software is a set of programs to carry out operations for a specific application. System software is a set of programs written for performing tasks such as controlling all operations required to move data into and out of the computer.
  3. (a) False      (b) True      (c) True
  4. FORTRAN is used for scientific applications whereas COBOL is used for business applications.
  5. Machine language is the only language that is directly understood by the computer. It is written in binary form that is 0 and 1. The set of symbols and letters forms the Assembly Language.
  6. (a) True      (b) False      (c) True
  7. Both compiler and interpreter are program translators used for translating higher-level language into machine language. While compiler scans the entire program first and then translates it into machine code, an Interpreter translates and executes the program line by line.
  8. FORTRAN (Formula Translation) and BASIC (Beginners All-Purpose Symbolic Instruction Code) are some of the high level languages.
  9. (a) True      (b) False      (c) True
-