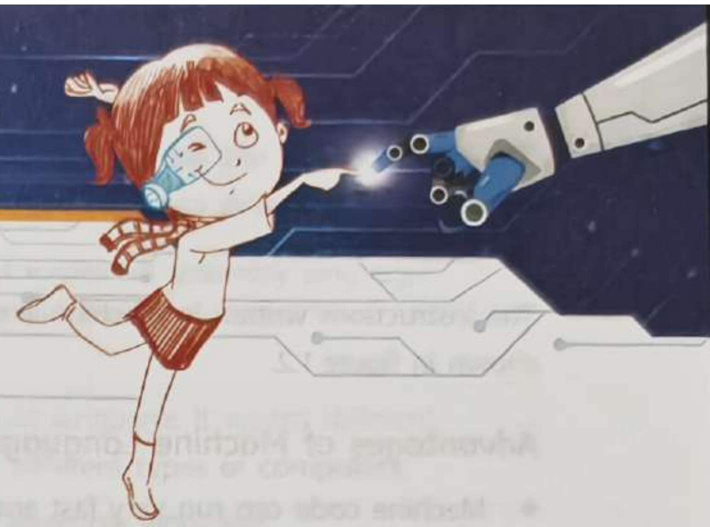

1

Computer Languages

Friends, in our country, every state has a different language. To speak to a person from a different state, you need to know his or her language or any other common language that both can understand. However, a computer does not understand any human language. Then, how do we communicate with it? Let us learn how!

LET'S LEARN ABOUT

- First Generation Language
- Second Generation Language
- Third Generation Languages
- Fourth Generation Languages
- Fifth Generation Languages
- Translators

A computer needs specific instructions to perform a task or function. These sets of instructions are called programs. The process of developing a set of instructions to enable a computer to do a particular task is called programming. Each programming language has its own set of rules.

Programming languages can be broadly classified into five categories (Fig. 1.1) on the basis of their development.

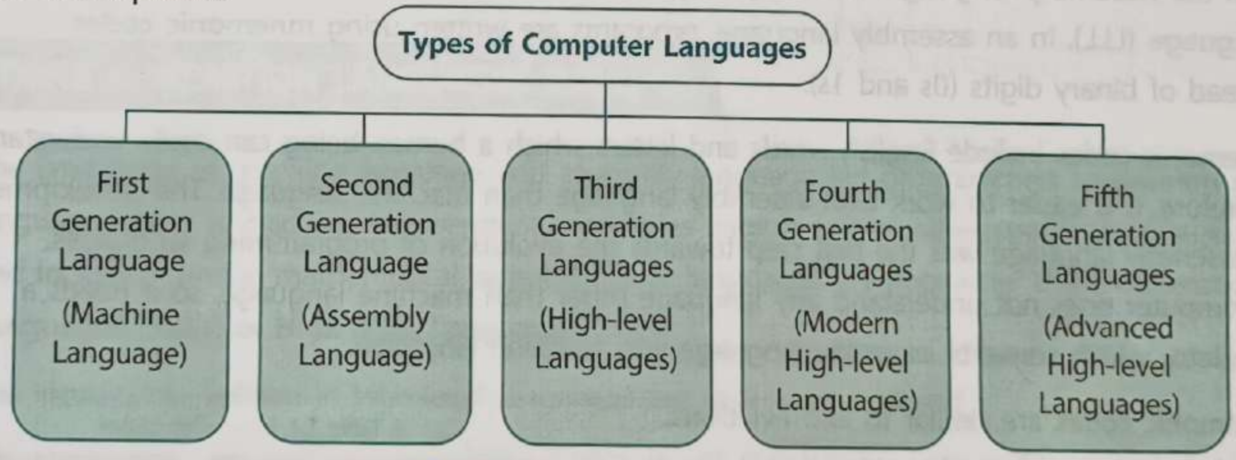
Types of Computer Languages
First Generation Language (Machine Language)
Second Generation Language (Assembly Language)
Third Generation Languages (High-level Languages)
Fourth Generation Languages (Modern High-level Languages)
Fifth Generation Languages (Advanced High-level Languages)

Fig. 1.1 Types of computer languages

FIRST GENERATION LANGUAGE (1GL)

First Generation Language (1GL) or Machine language is also known as Low-level Language (LLL). Machine language is the only language that a computer directly understands.

The instructions in 1GL are made of binary numbers, represented by 1s and 0s. Programs can then be executed directly in machine language.

The instructions written in machine language look like the one shown in figure 1.2.

Fig. 1.2 Machine language

Advantages of Machine Language

- Machine code can run very fast and efficiently since the instructions are executed directly by the CPU.
- It utilises less memory.

Disadvantages of Machine Language

- It is difficult to debug the programs written in machine language.
- It is extremely machine-dependent language. Hence, you can use only one type of computer for a particular machine language.

INFO BOX

The binary digit 0 represents the OFF state, and the binary digit 1 represents the ON state.

SECOND GENERATION LANGUAGE (2GL)

Second Generation Language (2GL) refers to the programming language that is associated with the Assembly language. Assembly language is also regarded as Low-level Language (LLL). In an assembly language, programs are written using mnemonic codes instead of binary digits (0s and 1s).

Mnemonic codes include English words and letters which a human being can easily understand. Therefore, it is easier to work with assembly language than machine language. The development of assembly language was the first step towards the evolution of programming languages. A computer does not understand any language other than machine language, so it needs a translator which converts assembly language into machine language.

Mnemonic codes are similar to abbreviations, which help in remembering the original words or concepts.

For example, to add two numbers 3 and 5, you can simply write a code as `ADD 3, 5` instead of giving a long binary code. Table 1.1 shows some common mnemonic codes.

Table 1.1 Mnemonic codes

Word	Mnemonic Code
Addition	ADD
Multiplication	MUL
Subtraction	SUB
Division	DIV

Advantages of Assembly Language

- It uses English words which make it readable by human beings.
- It is easy to locate and fix the errors in the programs written in assembly language.

Disadvantages of Assembly Language

- Like machine language, it is also a machine-dependent language. It means different assembly language programs need to be written for different types of computers.
- It is quite time-consuming to write the programs in assembly language.

LET'S REVIEW

Write T for True and F for False.

1. A computer works with the help of instructions. ______
2. Working with assembly language is difficult as compared to machine language. ______
3. Machine language is regarded as a high-level language. ______
4. ADD, SUB, and DIV are some examples of mnemonic codes. ______
5. The instructions in 1GL are made of binary numbers, represented by 1s and 0s. ______
6. Assembly language programs need to be written for different types of computers. ______

THIRD GENERATION LANGUAGES (3GLs)

The limitations of machine language and assembly language led programmers to develop a language which is machine independent, resembles English or any other spoken language, and includes familiar mathematical symbols. These languages are known as Third Generation Languages (3GLs) or High-level Languages (HLL).

The instructions written in high-level languages are easy to understand and write. Some examples of high-level languages are FORTRAN, COBOL, BASIC, Pascal, C, C++, Java, and so on.

Like assembly language, a high-level language also needs a translator which converts high-level language programs into the machine language. You will study about different types of translators later in the chapter.

INFO BOX

A translator is also known as language processor.

Features of High-level Language

- It is user-friendly.
- It uses English words and mathematical operators which make it easy to understand.
- It is not dependent on machine.

FOURTH GENERATION LANGUAGES (4GLs)

Fourth Generation Languages (4GLs) or Modern High-level Languages were developed to get closer to human language in terms of thinking and conceptualisation than any previous generation languages. It is independent of operating system.

Some examples of 4GLs are SQL, Focus, PostScript, WAVE, Python, Ruby, Perl, and so on.

Features of Fourth Generation Languages

- It is more programmer-friendly and versatile.
- It requires minimum efforts from the user to write the program.
- It enhances programming efficiency with the usage of English-like words and phrases, icons, graphical interfaces, and symbolic representation.
- It possesses high-speed of execution as compared to the languages of previous generations.
- The fourth generation languages are designed to reduce the overall time, effort, and cost that is incurred during the development of any software.

FIFTH GENERATION LANGUAGES (5GLs)

Fifth Generation Languages (5GLs) or Advanced High-level Languages are used in Artificial Intelligence research. Its primary goal is to eliminate the need of implementing an algorithm to solve a problem. It means that unlike the earlier generations of programming languages, which were designed to build specific programs, the fifth generation languages are designed to make the computer solve a given problem on its own without any interference of the programmer. Its ultimate aim is to make computers behave, think, and react as human beings.

Once a programmer dictates how the solution should appear by logically specifying constraints and conditions, the computer is then free to search for a suitable solution. A considerable research has gone into the development of 5GLs in the 1980s and 1990s. Fifth Generation programming languages are still in a development phase. It is believed that the fifth generation languages will lead the future.

Some examples of 5GLs are Prolog, OPS5, and Mercury.

TRANSLATORS

A translator or language processor converts the instructions into a machine language to make it a computer understandable language. Figure 1.3 shows different types of translators used by a computer.

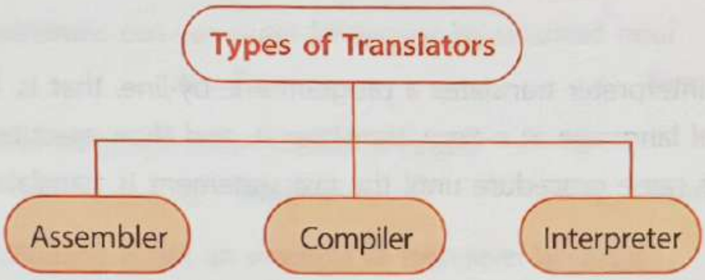

Fig. 1.3 Types of translators

Assembler

An Assembler translates the assembly language programs into machine language programs (Fig. 1.4), which can be understood by the processor.

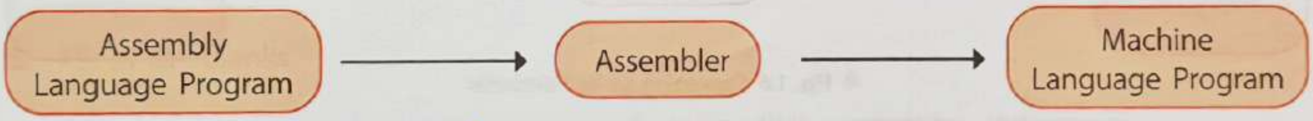

Fig. 1.4 Conversion by an assembler

INFO BOX

The program written in assembly language or high-level language is called Source Program or Source Code, and the program converted into machine language using a translator is known as Object Program or Object Code.

Compiler

A Compiler is a language translator program which converts a high-level language program into machine language (Fig. 1.5). It translates the entire program into machine language at once, as soon as all syntax errors are removed. It takes less time to compile source code.

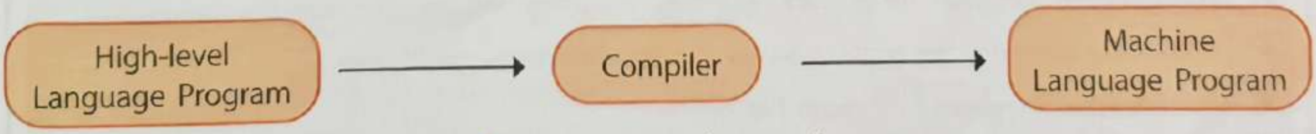

Fig. 1.5 Conversion by a compiler

INFO BOX

Compilers were introduced in 1952, since then hundreds of third generation languages (3GLs) have been developed, benefiting programmers in business and science.

Interpreter

Unlike a compiler, an interpreter translates a program line-by-line, that is, it takes up one statement of high-level language at a time, translates it, and then executes the instructions. Interpreter repeats the same procedure until the last statement is translated into machine language (Fig. 1.6).

If there are any syntax errors in the source program, they are brought to attention of the programmer as soon as a program statement is interpreted. Thus, it makes finding and correcting errors easier and quicker.

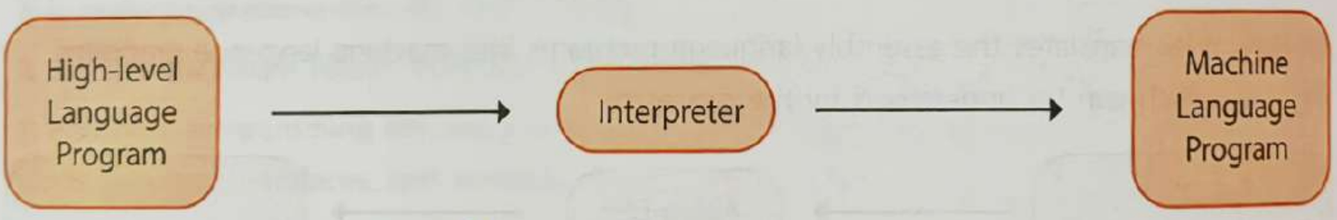

Fig. 1.6 Conversion by an interpreter

FLASHBACK

- The process of developing sets of instructions in a computer language is called programming.
- Machine language is the fundamental language of computers. It is the only language that computers directly understand.
- In an assembly language, mnemonic codes or symbols are used instead of binary digits (0s and 1s).
- High-level languages are considered as third generation languages.
- Fourth generation languages are more programmer-friendly and versatile.
- Fifth generation languages are a grouping of programming languages based on the fact that problems can be solved by providing constraints to the computer, rather than algorithmically specifying how the problem needs to be solved.
- An Assembler translates the assembly language programs into machine language programs.
- A Compiler translates the entire program into machine language at once.
- An Interpreter translates a program line-by-line.

EXERCISES

A. **Tick (✓) the correct answers.**

1. How many generations can computer languages be classified into?

   a. Two　　b. Four　　c. Five

2. This generation of language uses mnemonic codes.

   a. Second　　b. Third　　c. Fourth

3. Which of the following is not an example of high-level language?

   a. Machine　　b. COBOL　　c. Java

4. Which of the following does not translate the entire program at once?

   a. Interpreter　　b. Compiler　　c. Both a and b

5. In which year were the compilers introduced?

   a. 1992　　b. 2002　　c. 1952

B. **Fill in the blanks.**

1. ______________ language is the only language that a computer understands.

2. Assembly language is regarded as the ______________ generation language.

3. A program converted into machine language by a translator is called ______________

4. A/An ______________ converts a high-level language program into machine language, line-by-line.

5. ______________ is an example of fifth generation language.

C. **Write T for True and F for False.**

1. A program is a set of instructions. ____

2. The development of computer languages is classified into two categories. ____

3. Assembly language consists of binary numbers, 0s and 1s. ____

4. Machine language uses simple English words and phrases. ____

5. The ultimate aim of 5GLs is to make computers behave, think, and react as human beings. ____

D. **Answer the following questions.**

1. What do you understand by a programming language? Define machine language.

2. What are Second Generation Languages? List the advantages and disadvantages of assembly language.

3. Differentiate between source code and object code.

4. Define the term translator. How is an interpreter different from a compiler?

5. What are the characteristics of the fifth generation languages? Give some examples of fifth generation languages.