Chapter 2
Programming Languages
Topics

• Definition of Program, Computer Programming, and Computer Programmer.
• Generations of Programming Language
• Types of Programming Language
Computer Program

• A program is a set of instructions following the rules of the chosen language.
• Without programs, computers are useless.
• A program is like a recipe.
• It contains a list of ingredients (called variables) and a list of directions (called statements) that tell the computer what to do with the variables.
Programming Language

• A vocabulary and set of grammatical rules (syntax) for instructing a computer to perform specific tasks.
• Programming languages can be used to create computer programs.
• The term programming language usually refers to high-level languages, such as BASIC, C, C++, COBOL, FORTRAN, Ada, and Pascal.
Programming Language

• You eventually need to convert your program into machine language so that the computer can understand it.

• There are two ways to do this:
  – Compile the program
  – Interpret the program
Programming Language

- **Compile** is to transform a program written in a high-level programming language from source code into object code.
- This can be done by using a tool called **compiler**.
- A compiler reads the whole source code and translates it into a complete machine code program to perform the required tasks which is output as a new file.
Programming Language

- **Interpreter** is a program that executes instructions written in a high-level language.
- An interpreter reads the source code one instruction or line at a time, converts this line into machine code and executes it.
Computer Programming

• Computer programming is the process of writing, testing, debugging/troubleshooting, and maintaining the source code of computer programs.

• This source code is written in a programming language like C++, JAVA, Perl etc.
Computer Programmer

• A **programmer** is someone who writes computer program.
• Computer programmers write, test, and maintain programs or software that tell the computer what to do.
What Skills are Required to Become a Programmer?

• **Programming** - Writing computer programs for various purposes.

• **Writing** - Communicating effectively with others in writing as indicated by the needs of the audience.

• **Reading Comprehension** - Understanding written sentences and paragraphs in work-related documents.

• **Critical Thinking** - Using logic and analysis to identify the strengths and weaknesses of different approaches.
What Skills are Required to Become a Programmer?

- **Computers and Electronics** - Knowledge of electric circuit boards, processors, chips, and computer hardware and software, including applications and programming.

- **Mathematics** - Knowledge of numbers, their operations, and interrelationships including arithmetic, algebra, geometry, calculus, statistics, and their applications.

- **Oral Expression** - The ability to communicate information and ideas in speaking so others will understand.
What Skills are Required to Become a Programmer?

• **Oral Comprehension** - The ability to listen to and understand information and ideas presented through spoken words and sentences.

• **Written Expression** - The ability to communicate information and ideas in writing so others will understand.

• **Written Comprehension** - The ability to read and understand information and ideas presented in writing.
What Skills are Required to Become a Programmer?

• **Deductive Reasoning** - The ability to apply general rules to specific problems to come up with logical answers. It involves deciding if an answer makes sense.

• **Information Organization** - Finding ways to structure or classify multiple pieces of information.
Generations of Programming Language

- The **first generation languages**, or 1GL, are low-level languages that are machine language.
- The **second generation languages**, or 2GL, are also low-level languages that generally consist of assembly languages.
- The **third generation languages**, or 3GL, are high-level languages such as C.
Generations of Programming Language

- The **fourth generation languages**, or 4GL, are languages that consist of statements similar to statements in a human language. Fourth generation languages are commonly used in database programming and scripts.

- The **fifth generation languages**, or 5GL, are programming languages that contain visual tools to help develop a program. A good example of a fifth generation language is Visual Basic.
Types of Programming Language

- There are three types of programming language:
  - Machine language (Low-level language)
  - Assembly language (Low-level language)
  - High-level language

- Low-level languages are closer to the language used by a computer, while high-level languages are closer to human languages.
Machine Language

• Machine language is a collection of binary digits or bits that the computer reads and interprets.
• Machine languages are the only languages understood by computers.
• While easily understood by computers, machine languages are almost impossible for humans to use because they consist entirely of numbers.
Machine Language

Machine Language

169 1 160 0 153 0 128 153 0 129 153 130 153 0 131
200 208 241 96

High level language

5 FOR I=1 TO 1000: PRINT "A";: NEXT I
Machine Language

Example:

- Let us say that an electric toothbrush has a processor and main memory.
- The processor can rotate the bristles left and right, and can check the on/off switch.
- The machine instructions are one byte long, and correspond to the following machine operations:
### Machine Language

<table>
<thead>
<tr>
<th>Machine Instruction</th>
<th>Machine Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000 0000</td>
<td>Stop</td>
</tr>
<tr>
<td>0000 0001</td>
<td>Rotate bristles left</td>
</tr>
<tr>
<td>0000 0010</td>
<td>Rotate bristles right</td>
</tr>
<tr>
<td>0000 0100</td>
<td>Go back to start of program</td>
</tr>
<tr>
<td>0000 1000</td>
<td>Skip next instruction if switch is off</td>
</tr>
</tbody>
</table>
Assembly Language

• A program written in assembly language consists of a series of instructions mnemonics that correspond to a stream of executable instructions, when translated by an assembler, that can be loaded into memory and executed.

• Assembly languages use keywords and symbols, much like English, to form a programming language but at the same time introduce a new problem.
Assembly Language

• The problem is that the computer doesn't understand the assembly code, so we need a way to convert it to machine code, which the computer does understand.

• Assembly language programs are translated into machine language by a program called an assembler.
Assembly Language

- Example:
  - Machine language :
    10110000 01100001
  - Assembly language :
    \texttt{mov a1, \#061h}
  - Meaning:
    Move the hexadecimal value 61 (97 decimal) into the processor register named "a1".
High Level Language

- **High-level** languages allow us to write computer code using instructions resembling everyday spoken language (for example: `print`, `if`, `while`) which are then translated into machine language to be executed.

- Programs written in a high-level language need to be translated into **machine language** before they can be executed.

- Some programming languages use a **compiler** to perform this translation and others use an **interpreter**.
High-Level Language

• Examples of High-level Language:
  • ADA
  • C
  • C++
  • JAVA
  • BASIC
  • COBOL
  • PASCAL
  • PHYTON
## Comparisson

<table>
<thead>
<tr>
<th></th>
<th>Machine Language</th>
<th>Assembly Language</th>
<th>High-level Languages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time to execute</td>
<td>Since it is the basic language of the computer, it does not require any translation, and hence ensures better machine efficiency. This means the programs run faster.</td>
<td>A program called an ‘assembler’ is required to convert the program into machine language. Thus, it takes longer to execute than a machine language program.</td>
<td>A program called a compiler or interpreter is required to convert the program into machine language. Thus, it takes more time for a computer to execute.</td>
</tr>
<tr>
<td>Time to develop</td>
<td>Needs a lot of skill, as instructions are very lengthy and complex. Thus, it takes more time to program.</td>
<td>Simpler to use than machine language, though instruction codes must be memorized. It takes less time to develop programs as compared to machine language.</td>
<td>Easiest to use. Takes less time to develop programs and, hence, ensures better program efficiency.</td>
</tr>
</tbody>
</table>
BASIC

• Short for **Beginner's All-purpose Symbolic Instruction Code**.
• Developed in the 1950s for teaching University students to program and provided with every self-respecting personal computer in the 1980s,
• BASIC has been the first programming language for many programmers.
• It is also the foundation for Visual Basic.
BASIC

Example:

PRINT "Hello world!"
Visual Basic

- A programming language and environment developed by Microsoft.
- Based on the BASIC language, Visual Basic was one of the first products to provide a graphical programming environment and a paint metaphor for developing user interfaces.
Visual Basic

Example:

MsgBox "Hello, World!“
C

- Developed by Dennis Ritchie at Bell Labs in the mid 1970s.
- C is much closer to assembly language than are most other high-level languages.
- The first major program written in C was the UNIX operating system.
- The low-level nature of C, however, can make the language difficult to use for some types of applications.
C

Example:

```c
#include <stdio.h>

int main(void)
{
    printf("hello, world\n");
    return 0;
}
```
C++

• A high-level programming language developed by Bjarne Stroustrup at Bell Labs.
• C++ adds object-oriented features to its predecessor, C.
• C++ is one of the most popular programming language for graphical applications, such as those that run in Windows and Macintosh environments.
C++

Example:

```cpp
#include <iostream>

int main()
{
    std::cout << "Hello World!" << std::endl;
    return 0;
}
```
Pascal

- A high-level programming language developed by Niklaus Wirth in the late 1960s.
- The language is named after Blaise Pascal, a seventeenth-century French mathematician who constructed one of the first mechanical adding machines.
- It is a popular teaching language.
Pascal

Example:

```pascal
Program HelloWorld(output);
begin
  writeLn('Hello, World!')
end.
```
Java

- A high-level programming language developed by Sun Microsystems.
- Java was originally called OAK, and was designed for handheld devices and set-top boxes.
- Oak was unsuccessful so in 1995 Sun changed the name to Java and modified the language to take advantage of the burgeoning World Wide Web.
- Java is a general purpose programming language with a number of features that make the language well suited for use on the World Wide Web.
Java

Example:

/* * Outputs "Hello, World!" and then exits */

public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}

Choosing a Programming Language

Before you decide on what language to use, you should consider the following:

• your server platform
• the server software you run
• your budget
• previous experience in programming
• the database you have chosen for your backend