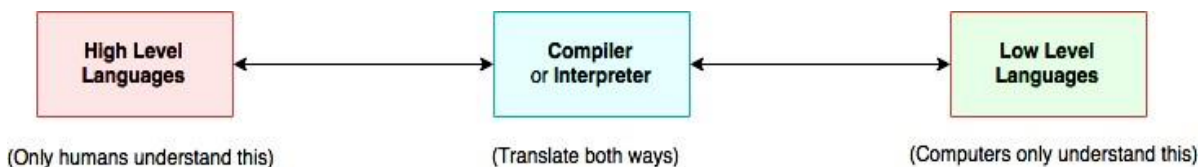
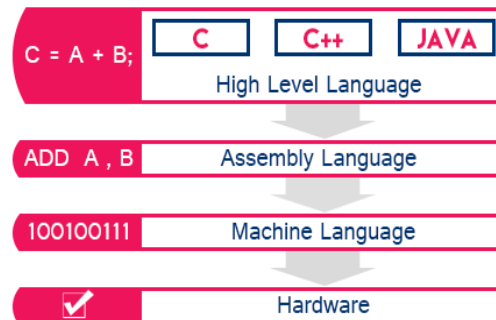
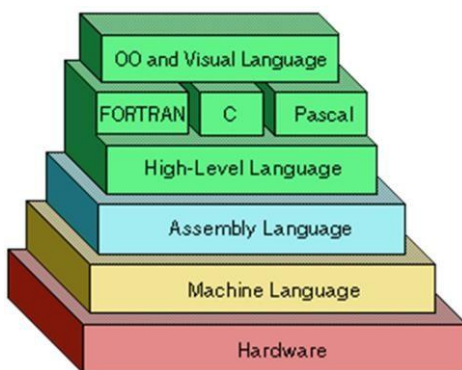
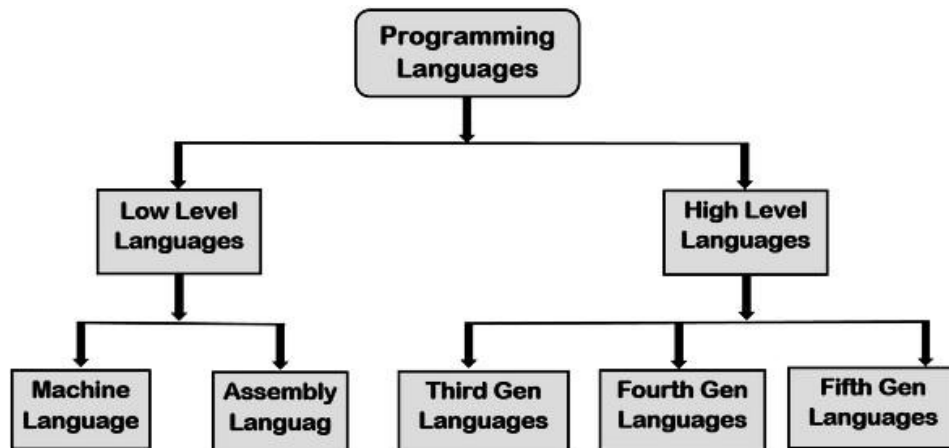


## Programming Languages

*Computer programs, known as software, are instructions that tell a computer what to do. Computers do not understand human languages, so programs must be written in a language a computer can use. All programs must be converted into the instructions the computer can execute.*

### Programming Language Type



## Machine Language

A computer's native language, which differs among different types of computers, is its *machine language*—a set of built-in primitive instructions. These instructions are in the form of binary code. For example, to add two numbers, you might have to write an instruction in binary code,

like this:

1101101010011010

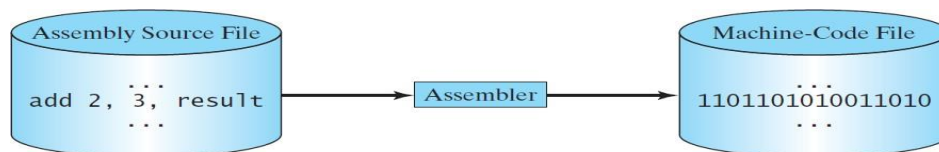
## Assembly Language

Programs written in machine language are very difficult to read and modify. For this reason, *assembly language* was created in the early days of computing as an alternative to machine languages. Assembly language uses a short descriptive word, known as a *mnemonic*, to represent each of the machine-language instructions.

For example, the mnemonic `add` typically means to add numbers and `sub` means to subtract numbers. To add the numbers 2 and 3 and get the result, you might write an instruction in assembly code like this:

```
add 2, 3, result
```

*assembler*—is used to translate assembly-language programs into machine code, as shown in Figure 1.3.



**FIGURE 1.3** An assembler translates assembly-language instructions into machine code.

## High-Level Language

High-level languages are English-like and easy to learn and use. The instructions in a high-level programming language are called *statements*.

For example, is a high-level language statement that computes the area of a circle with a radius 5:

```
area = 5 * 5 * 3.14;
```

There are many high-level programming languages, and each was designed for a specific purpose. Table 1.1 lists some popular ones.

**TABLE 1.1** Popular High-Level Programming Languages

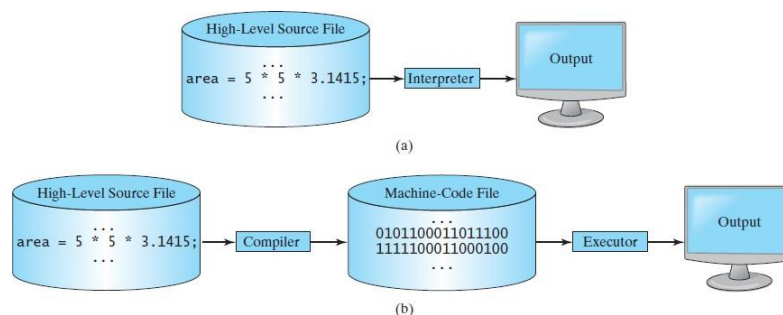
Language	Description
Ada	Named for Ada Lovelace, who worked on mechanical general-purpose computers. The Ada language was developed for the Department of Defense and is used mainly in defense projects.
BASIC	Beginner's All-purpose Symbolic Instruction Code. It was designed to be learned and used easily by beginners.
C	Developed at Bell Laboratories. C combines the power of an assembly language with the ease of use and portability of a high-level language.
C++	C++ is an object-oriented language, based on C.
C#	Pronounced "C Sharp." It is a hybrid of Java and C++ and was developed by Microsoft.
COBOL	COMmon Business Oriented Language. Used for business applications.
FORTRAN	FORmula TRANslation. Popular for scientific and mathematical applications.
Java	Developed by Sun Microsystems, now part of Oracle. It is widely used for developing platform-independent Internet applications.
Pascal	Named for Blaise Pascal, who pioneered calculating machines in the seventeenth century. It is a simple, structured, general-purpose language primarily for teaching programming.
Python	A simple general-purpose scripting language good for writing short programs.
Visual Basic	Visual Basic was developed by Microsoft and it enables the programmers to rapidly develop graphical user interfaces.

A program written in a high-level language is called a *source program* or *source code*. A source program must be translated into machine code for execution. The translation can be done using another programming tool called an *interpreter* or a *compiler*.

- An interpreter reads one statement from the source code, translates it to the machine code or virtual machine code, and then executes it right away, as shown in Figure 1.4a.

Note that a statement from the source code may be translated into several machine instructions.

- A compiler translates the entire source code into a machine-code file, and the machine-code file is then executed, as shown in Figure 1.4b.



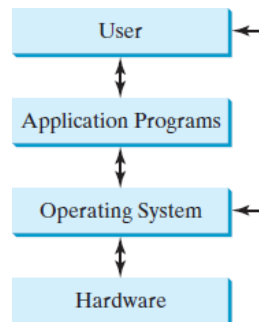
**FIGURE 1.4** (a) An interpreter translates and executes a program one statement at a time. (b) A compiler translates the entire source program into a machine-language file for execution.

## Operating Systems

The operating system (OS) is the most important program that runs on a computer.

The OS manages and controls a computer's activities.

The popular operating systems for general-purpose computers are Microsoft Windows, Mac, and Linux.



**FIGURE 1.5** Users and applications access the computer's hardware via the operating system.

The major tasks of an operating system are as follows:

- Controlling and monitoring system activities
- Allocating and assigning system resources
- Scheduling operations

## Programming Paradigms

### 1. Procedural Programming:

Procedural Programming can be defined as a programming model which is derived from structured programming, based upon the concept of calling procedure. Procedures, also known as routines, subroutines or functions, simply consist of a series of computational steps to be carried out.

#### **Languages used in Procedural Programming:**

FORTRAN, ALGOL, COBOL, BASIC, Pascal and C.

## 2. Object Oriented Programming:

**Object oriented programming** can be defined as a programming model which is based upon the concept of objects. Objects contain data in the form of attributes and code in the form of methods. In object oriented programming, computer programs are designed using the concept of objects that interact with real world. Object oriented programming languages are various but the most popular ones are class-based, meaning that objects are instances of classes, which also determine their types.

### Languages used in Object Oriented Programming:

Java, C++, C#, Python, PHP, JavaScript, Ruby, Perl, Objective-C, Dart, Swift.

PROCEDURAL PROGRAMMING	OBJECT ORIENTED PROGRAMMING
In procedural programming, program is divided into small parts called <b>functions</b> .	In object oriented programming, program is divided into small parts called <b>objects</b> .
Procedural programming follows <b>top down approach</b> .	Object oriented programming follows <b>bottom up approach</b> .
There is no access specifier in procedural programming.	Object oriented programming have access specifiers like private, public, protected etc.
Adding new data and function is not easy.	Adding new data and function is easy.
Procedural programming does not have any proper way for hiding data so it is <b>less secure</b> .	Object oriented programming provides data hiding so it is <b>more secure</b> .
In procedural programming, overloading is not possible.	Overloading is possible in object oriented programming.
In procedural programming, function is more important than data.	In object oriented programming, data is more important than function.
Procedural programming is based on <b>unreal world</b> .	Object oriented programming is based on <b>real world</b> .
Examples: C, FORTRAN, Pascal, Basic etc.	Examples: C++, Java, Python, C# etc.

## PROCEDURAL PROGRAMMING EXAMPLE

```
#include <iostream> using
namespace std;
int main()
{
    int i, fact = 1, num;
    cout << "Enter any Number: ";
    cin >> number;

    for (i = 1; i <= num; i++) {
        fact = fact * i;
    }
    cout << "Factorial of " << num << " is: " << fact <<
endl;
return 0;
}
```

## OBJECT ORIENTED PROGRAMMING EXAMPLE

```
class GFG {
    public static void main(String[] args)
    {
        System.out.println("GfG!");
        Signup s1 = new Signup();
        s1.create(22, "riya", "riya2@gmail.com", 'F', 89002);
    }
}
```

```
class Signup {
    int userid;
    String name;
    String emailid;
    char sex;
    long mob;

    public void create(int userid, String name,
        String emailid, char sex, long mob)
    {
        System.out.println("Welcome to
            GeeksforGeeks\nLets create your account\n");
        this.userid = 132;
        this.name = "Radha";
        this.emailid = "radha.89@gmail.com";
        this.sex = 'F';
        this.mob = 900558981;
        System.out.println("your account has been created");
    }
}
```

## **The Java Language Specification, API, JDK, and IDE**

- **Java syntax is defined in the Java language specification.**
- **The Application Program Interface (API), also known as library, contains predefined classes and interfaces for developing Java programs.**
- **The JDK is the software for developing and running Java programs.**
- **The IDE is an integrated development environment for rapidly developing programs like Eclipse, NetBeans.**