

Chapter Four Software Computer

For first stage

Prepared by
Wala'a N. Jasim

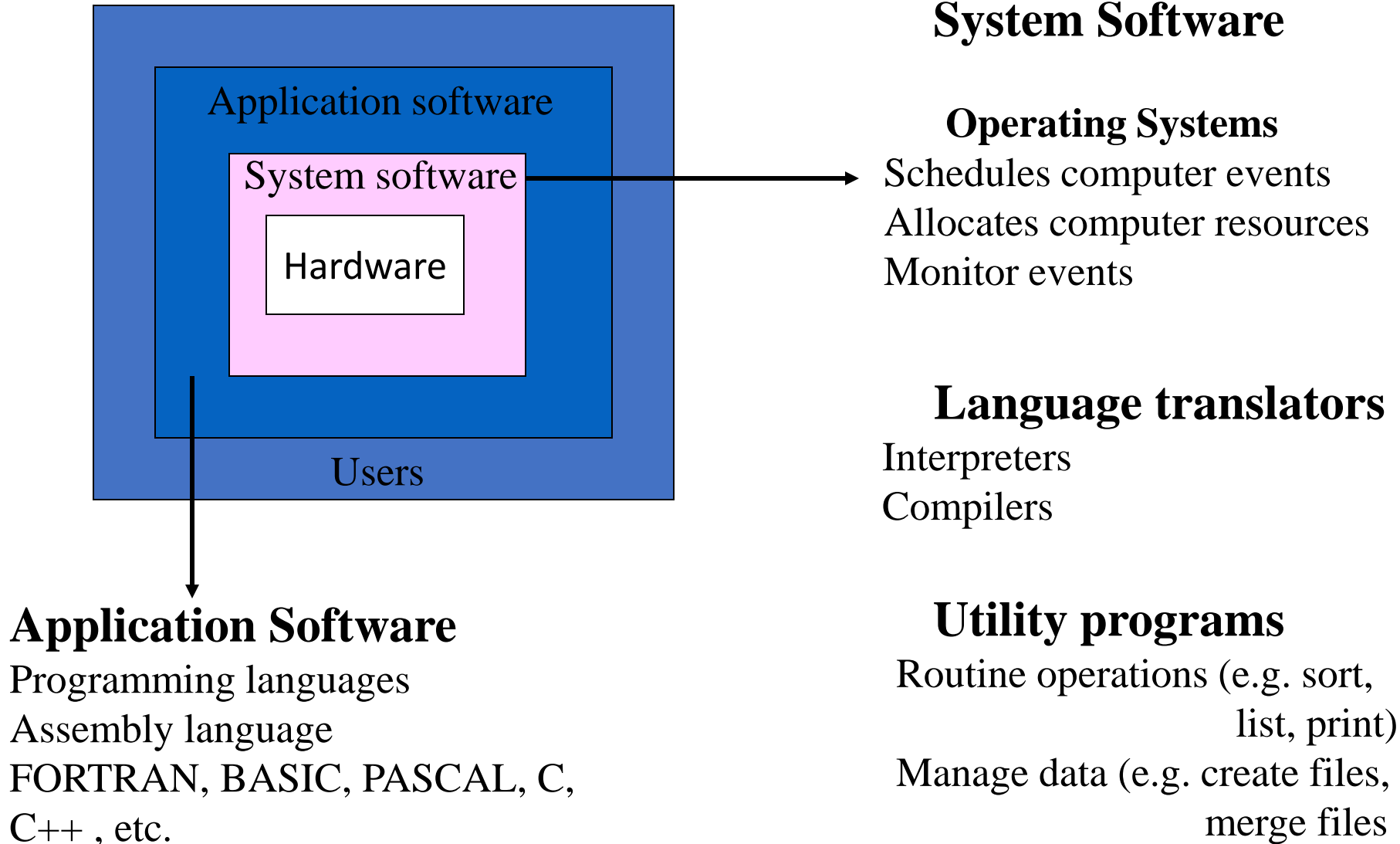
Software

- ❑ Hardware and Software have a **symbiotic relationship**. They need each other to fulfill their potential.
- ❑ Computer hardware is useless without software.
- ❑ Software is the set of instructions and associated data that direct the computer to do a task.
- ❑ Any set of instructions that guides the hardware and tells it how to accomplish each task.

Software

- ❑ Software can be divided into two categories:
system software and application software.
- ❑ System software helps the computer to carry out its basic operating tasks.
- ❑ Application software helps the user carry out a variety of tasks.

The major types of software



Software

System Software

- ❑ Manages the fundamental operations of the computer, such as
 - loading programs and data into memory
 - executing programs
 - saving data to disks
 - displaying information on the monitor
 - transmitting data through a port to a peripheral device.
- ❑ System software: operating systems, utilities, device drivers.

Operating System

Operating System

- Collection of computer programs that control the interaction of the user and the computer hardware.
- Responsible for directing all computer operations and managing all computer resources.
- Controls basic input and output, allocates system resources, manages storage space, maintains security, and detects equipment failure.

Operating System

- A part of the operating system code is stored in a ROM and the rest of it resides on a disk.
- Loading the operating system into memory is called booting the computer.

Operating System

Operating System Functions

- 1- Perform common computer hardware functions
- 2- Provide a user interface
- 3- Provide a degree of hardware independence
- 4- Manage system memory
- 5- Manage processing tasks
- 6- Provide networking capability
- 7- Control access to system resources
- 8- Manage files

Operating System

User Interface

A function of the operating system that allows individuals to access and command the computer

1- Command-based user interface

A particular user interface that requires text commands be given to the computer to perform basic activities

E.g., unix, DOS

Operating System

2-Graphical user interface (GUI)

A user interface that uses pictures (icons) and menus displayed on the screen to send commands to the computer system

E.g. Windows, MAC OS

Operating System

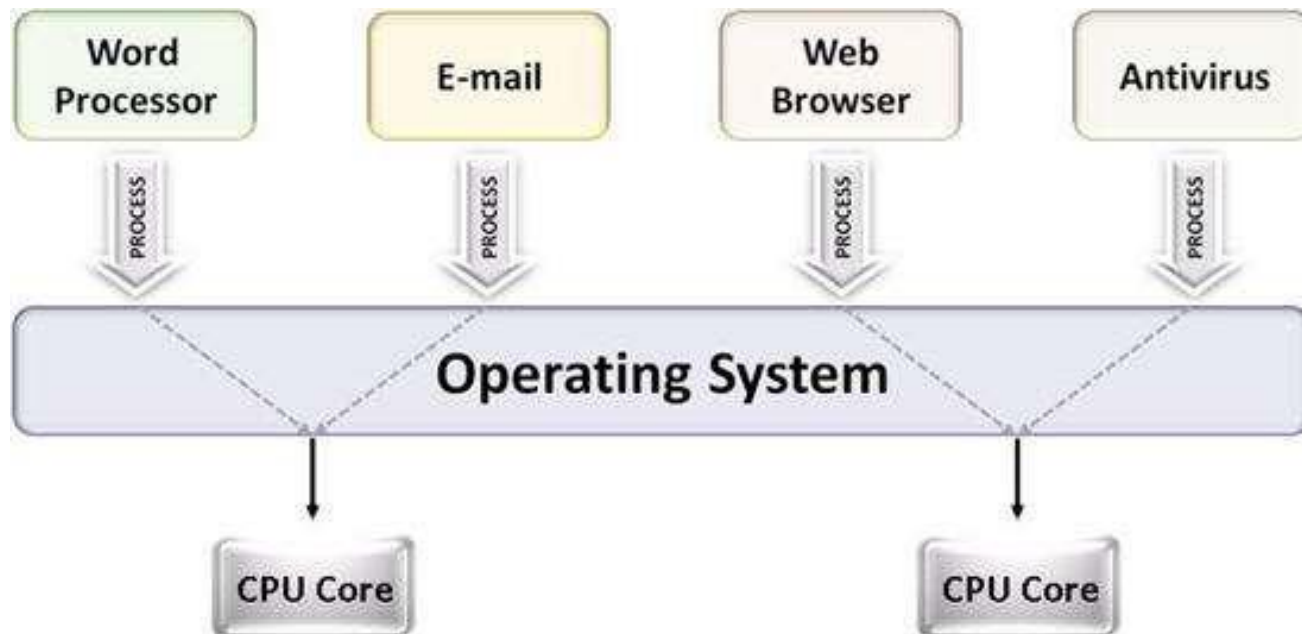
Operating System Types

1- Multi – Tasking Operating System:

- **Multitasking** : The multiprogramming capability of primarily single-user operating systems (PC).
- One person can run two or more programs concurrently on a single computer.
- When you are writing a report using MS Word, you can also search on the Internet.

Operating System

- Multitasking allows you to display both programs on the computer screen and work with them at the same time.

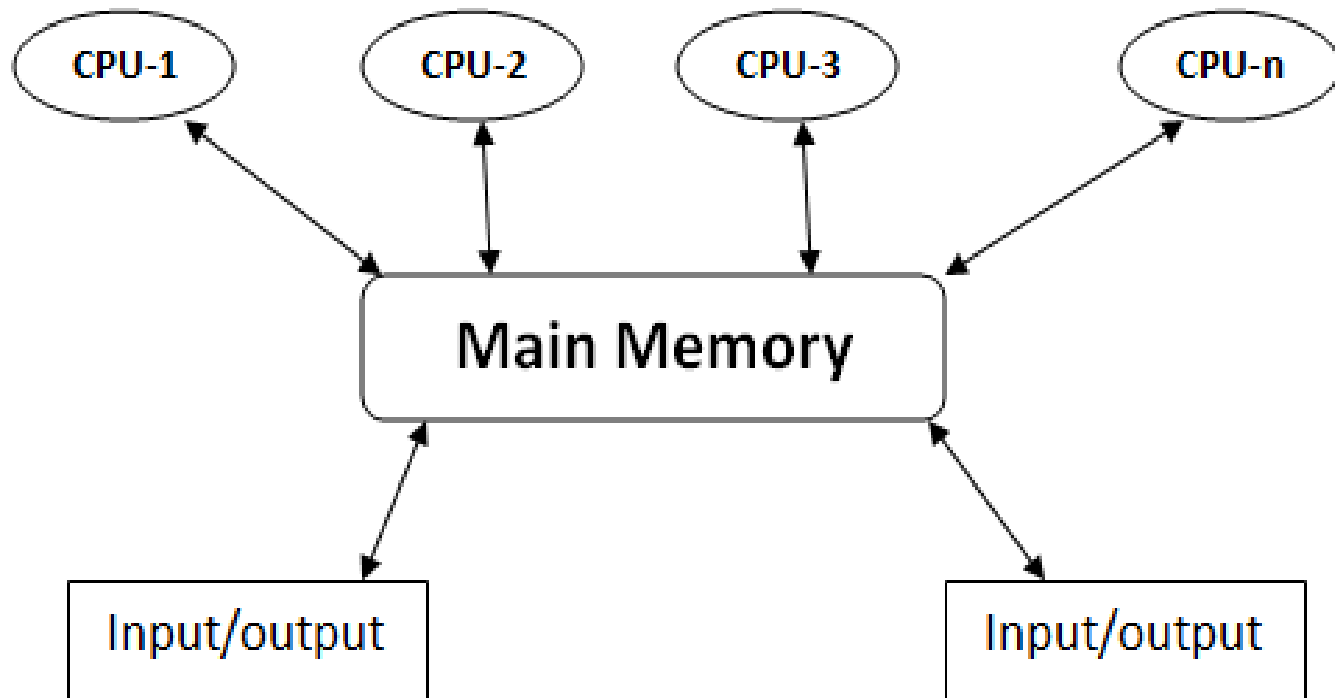


Operating System

2- Multi – Processing Operating System:

- **Multiprocessing** : an operating system capability that links together two or more CPU to work in parallel in a single computer system.
- The operating system can assign multiple CPUs to execute different instructions from the same program or from different programs at one time, dividing the work between the CPUs.
- Multiprocessing uses simultaneous processing with multiple CPUs.

Operating System



Operating System

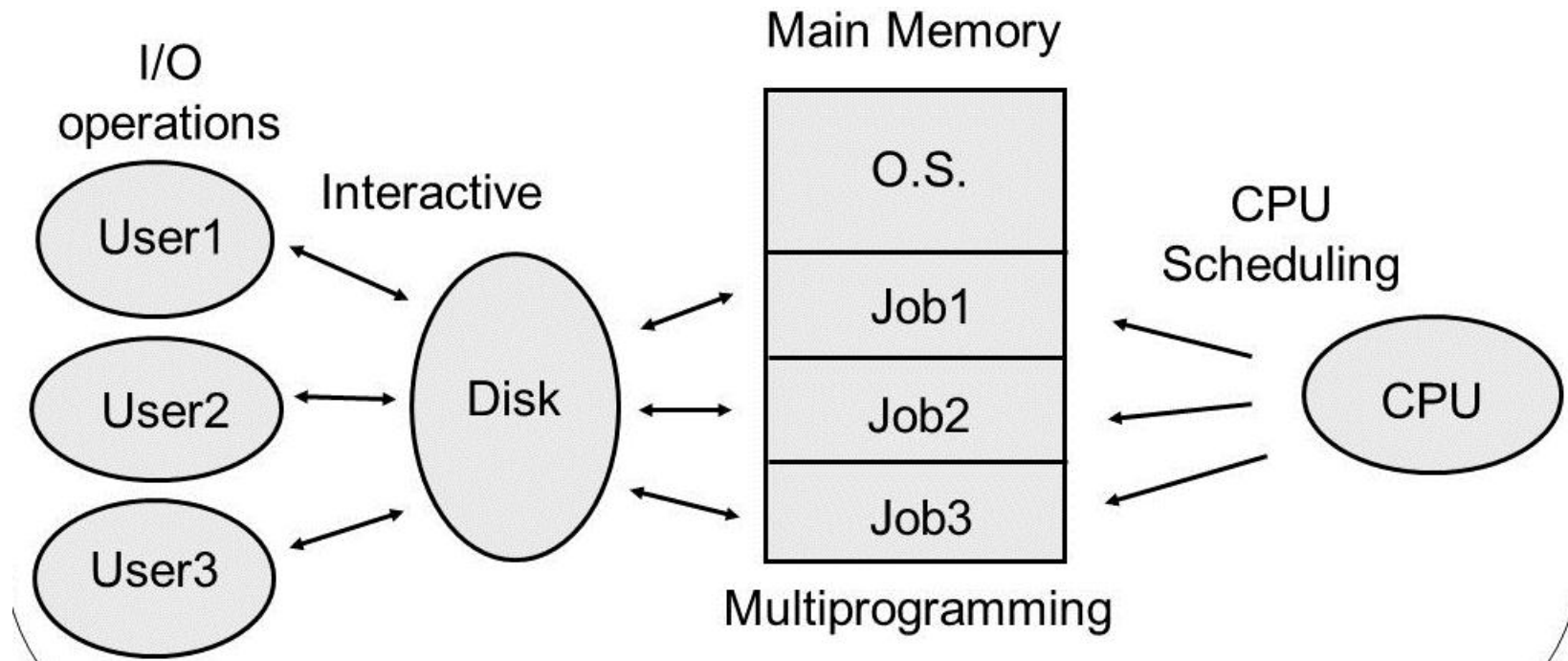
3- Time sharing Operating System:

- Time-sharing is a technique which enables many people, located at various terminals, to use a particular computer system at the same time.
- Time-sharing or multitasking is a logical extension of multiprogramming. Processor's time which is shared among multiple users simultaneously is termed as **time-sharing**.

Operating System

- The time sharing systems were developed to provide an interactive use of the computer system.
- A time shared system uses CPU scheduling and multiprogramming to provide each user with a small portion of a time-shared computer.
- It allows many users to share the computer resources simultaneously.

Operating System

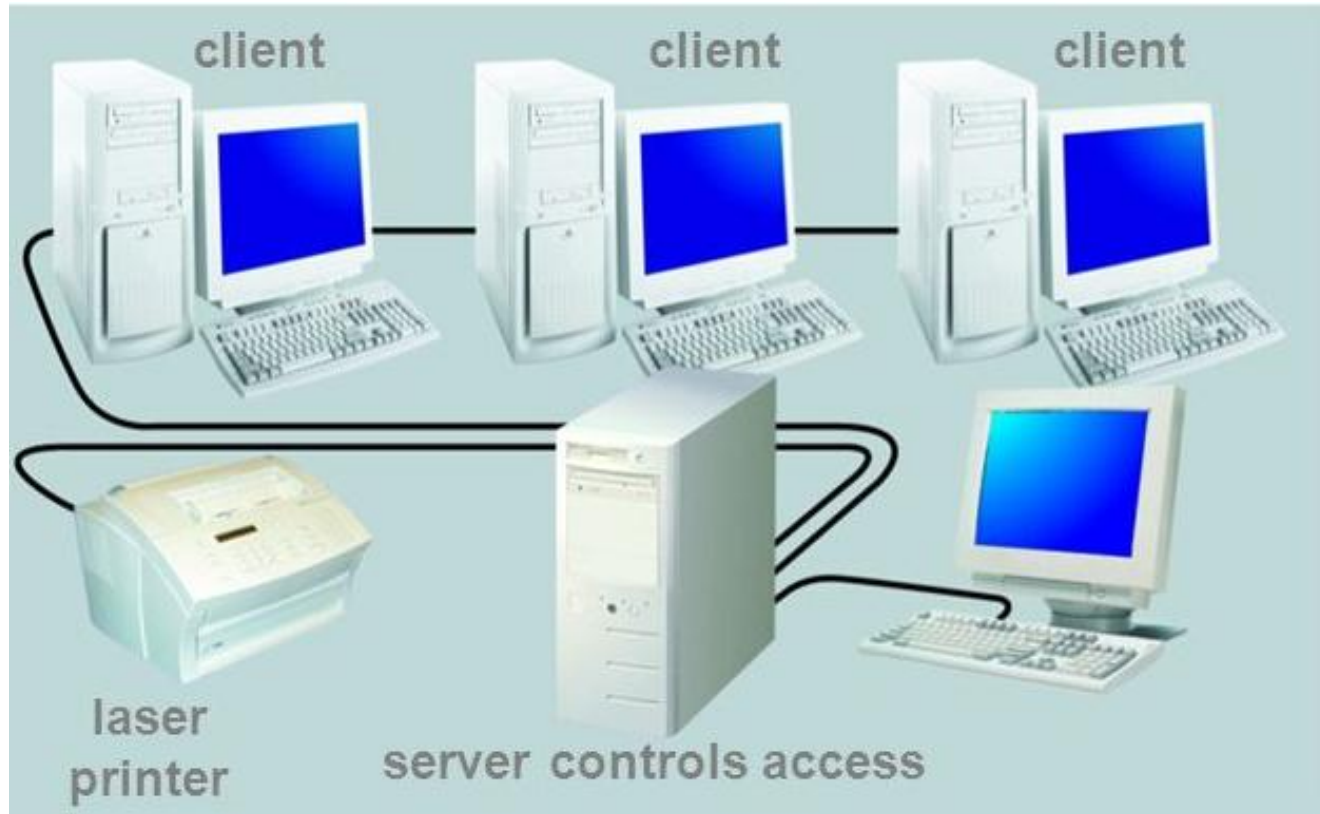


Operating System

4- Network Operating System:

- An operating system that supports a network .
- Also called a network OS or NOS.
- A network is a collection of computers and devices connected together via communications media and devices.
- Refers to software that implements an operating system of some kind that is oriented to computer networking.

Operating System



Operating System

5- A real-time operating system (RTOS)

- any operating system (OS) intended to serve real-time applications that process data as it comes in, typically without buffer delays.

System Support Prpgrams

System Support Prpgrams

- The second major category of systems software, system support programs, supports the operations, management, and users of a computer system by providing a variety of support services.
- Examples of system support programs are :
 - ✓ **System utilities**
 - ✓ **System performance monitors**
 - ✓ **System security monitors**

System Support Prpgrams

1- System utilities are programs that have been written to accomplish common tasks such as :

- sorting records
- checking the integrity of disk(i.e., amount of storage, available and existence of any damage).
- creating directories and subdirectories.

System Support Prpgrams

2- System performance monitors are programs that monitor the processing of jobs on a computer system. They monitor computer system performance and produce reports containing detailed statistics relating to the use of system resources, such as :

- processor time
- memory space
- input/output devices

System Support Prpgrams

3- System Security Monitors are programs that monitor the use of a computer system to protect it and its resources from unauthorized use or destruction. Such programs provide the computer security needed to allow only authorized users access to the system.

Application Software

Application software (app.)

- Application software consists of programs designed to make users more productive and/or assist with personal tasks.
 - ✓ To make business activities more efficient
 - ✓ To assist with graphics and multimedia projects .
 - ✓ To support home, personal, and educational tasks.
 - ✓ To facilitate communications

Application Software

Application Software include :

- Productivity software like Microsoft Office
- Specialty software
- Entertainment software
- Educational and reference Software
- Personal software

Application Software

Examples of Application Software :

Productivity software : Programs that enable you to perform tasks required in home, school, and business such as :

- Word processing program
- Spreadsheet programs
- Presentation programs
- Database programs



Application Software

Entertainment Software : Designed to provide users with entertainment .

- ❑ Categories include
 - Action and Adventure
 - Driving
 - Puzzles
 - Role-playing
 - Simulation

Application Software

Educational and reference Software :

- **Educational Software :**

Refers to the variety of software applications on the market that offer some form of instruction or training.

- **reference software**

Refer to the software applications that act as sources for reference materials, such as the standard atlases, dictionaries, and thesauri,

Application Software

Specialized Business Software :

Design to the needs of a particular company or industry .

Personal software :

Used to for tax preparation, financial planning, and personal accounting such as turbo-tax
Microsoft money and peachtree accounting

Freeware and Shareware

- **Freeware** is any copyrighted software that you can use for free.
- **Shareware** software that allows users to run it for a limited time free of charge

Programming Languages

Computer Program

- Is a set of sequence instruction that tell the computer what to do .
- Theses instructions follow the rules of the chosen language.
- Without programs, computers are useless.
- It contains a list of components (called variables) and a list of directions (called statements) that tell the computer what to do with the variables.

Programming Languages

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 Languages

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

There are two ways to do this:

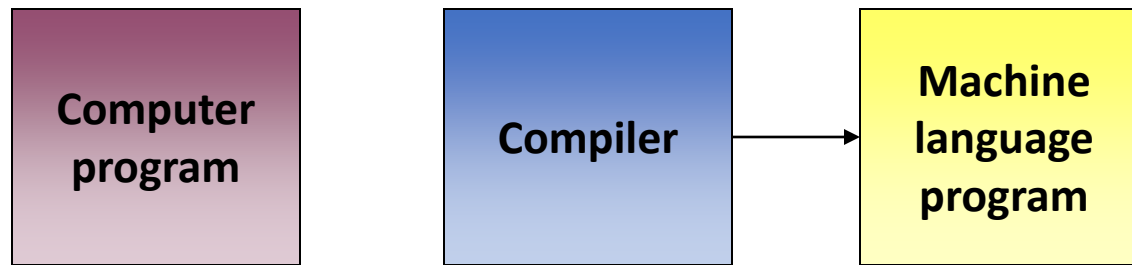
- –Compile the program
- –Interpret the program

Programming Languages

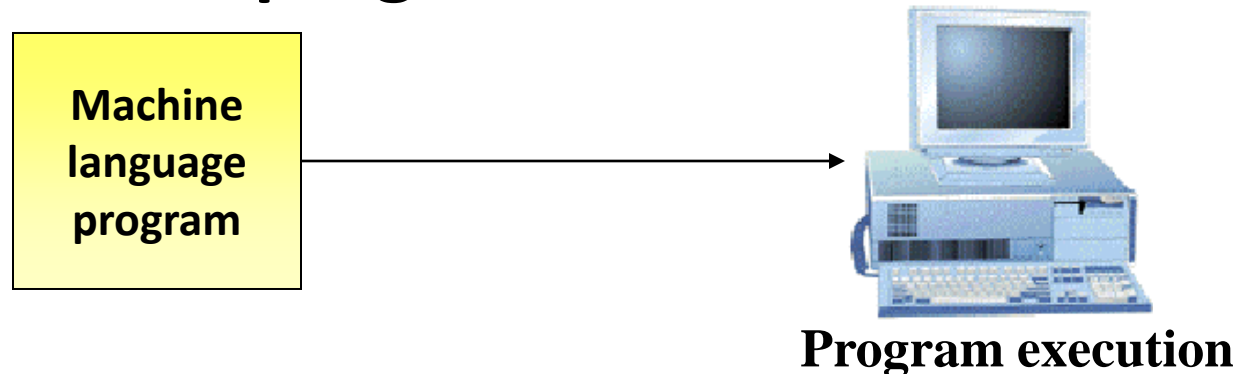
- **Compile** is to transform a program written in a high level programming language from source code into object code.
- This can be done by using a tool called **compiler**.
- A compiler reads the whole source code and translates it into a complete machine code program to perform the required tasks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

Programming Languages

Step 1: Translate program



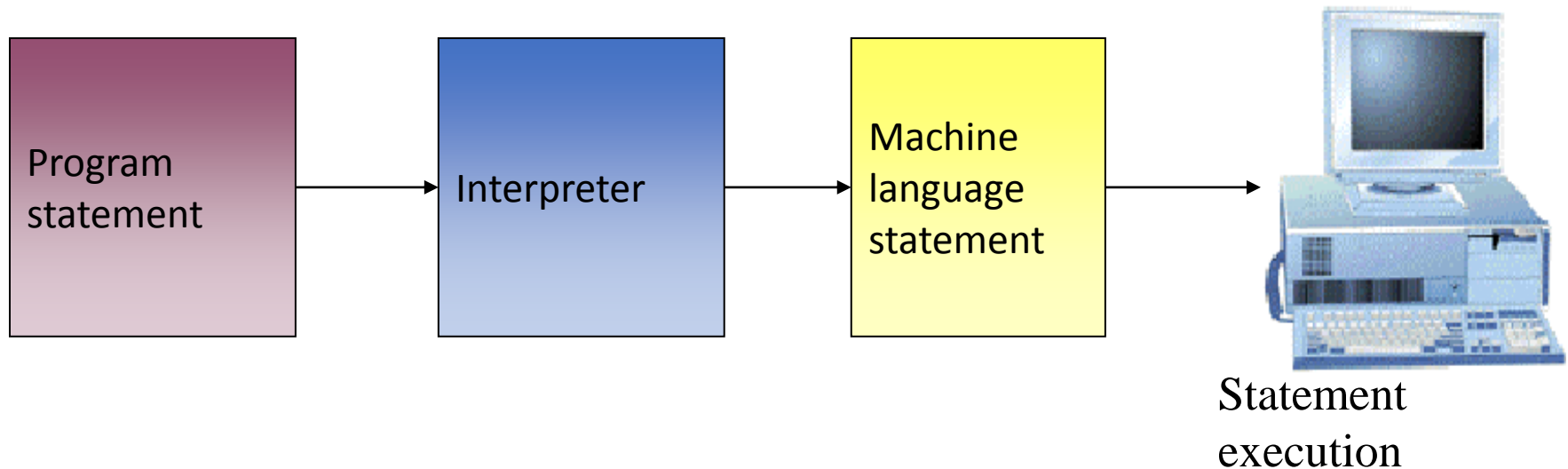
Step 2: Execute program



Programming Languages

- **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.

Programming Languages



Programming Languages

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 .

Programming Languages

Generations of Programming Language

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

Programming Languages

4- The fourth generation languages (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.

5- The fifth generation languages (5GL) are programming languages that contain visual tools to help develop a program. A good example of a fifth generation language is Visual Basic

Programming Languages

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.

Programming 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 .

Programming Languages

Example :

The machine instructions are one byte long and correspond to the following machine operations:

Machine Instruction	Machine Operation
0000 0000	Stop
0000 0001	Rotate bristles left
0000 0010	Rotate bristles right
0000 0100	Go back to start of program
0000 1000	Skip next instruction if switch is off

Programming Languages

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.

Programming Languages

- 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**.

Programming Languages

Example:

- ✓ Machine language :
1011000001100001
- ✓ Assembly language :
mov a1, #061h

Meaning:

Move the hexadecimal value 61 into the processor register named "a1".

Programming Languages

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

Programming Languages

- Some programming languages use a compiler to perform this translation and others use an interpreter.
- **Examples** of High-level Language:
C , C++ , JAVA ,BASIC , COBOL ,PASCAL .

Programming Languages

Comparison

	Machine Language	Assembly Language	High-level Languages
Time to execute	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.	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.	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.
Time to develop	Needs a lot of skill, as instructions are very lengthy and complex. Thus, it takes more time to program.	Simpler to use than machine language, though instruction codes must be memorized. It takes less time to develop programs as compared to machine language.	Easiest to use. Takes less time to develop programs and, hence, ensures better program efficiency.

The background of the slide features abstract geometric shapes, primarily triangles, in shades of blue and red. These shapes are layered and overlap, creating a modern, geometric aesthetic. The central text is positioned within the white space of the slide.

Thanks for your attention