Computer Software

Contents of Lecture:

- Computer Software
- Application and System Software
- Important Definitions
- Machine and Assembly Languages
- High-level Programming Languages

Computer Software

- **Software** is computer programs that run on a computer. Software is instructions that can be stored and run by hardware.

- Another term for a program or a set of programs is **software**, and we use both terms interchangeably throughout the text.

- Hardware is directed by the software to execute any command or instruction.
- A combination of hardware and software forms a usable computing system.

Application and System Software

- There are two types of computer programs are:
  - ✓ System software.
  - ✓ Application software

System software

- **System software** is the collection of programs that must be available to any computer system for it to operate.

- The most important system software is the **operating system**.
- Examples of some well-known operating systems include MS-DOS, UNIX, and MS WINDOWS.

- Many operating systems allow user to run multiple programs. Such operating systems are called **multitasking systems**. Beside operating systems, language translators are also system software.

Application software

- Application software consists of those programs written to perform particular tasks required by the users.
- Some Application software:





Important Definitions:

- **A computer program** is a set of instructions used to operate a computer to produce a specific result.

- These computer programs guide the computer through orderly sets of actions specified by person called **computer programmers**.

- **Programming languages** is a languages used to create computer programs.
- **Computer programming** mean writing a computer programs by a programmer using programming language.

Machine and Assembly Languages

Machine languages:

- Machine languages are the lowest level of computer languages. Programs written in machine language consist of entirely of 1s and 0s.

- Programs in machine language can control directly to the computer's hardware.

  00101010 000000000001 000000000010
  10011001 000000000010 000000000011

- A machine language instruction consists of two parts:
  - ✓ An instruction part
  - ✓ An address part.

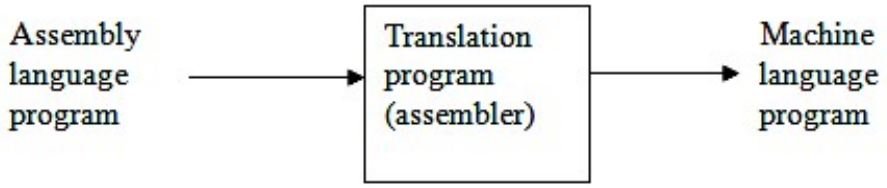
- The **instruction part** (opcode) is the leftmost group of bits in the instruction and tells the computer the operation to be performed.
- The **address part** specifies the memory address of the data to be used in the instruction.

Assembly languages:

- **Assembly languages** perform the same tasks as machine languages, but use symbolic names for (opcode) and operands instead of 1s and 0s.

```
LOAD BASEPAY
ADD OVERPAY
STORE GROSSPAY
```

- Since computers can only execute machine language programs, an assembly language program must be translated into a machine language program before it can be executed on a computer.



Assembly translation

- Machine languages and assembly languages are called low-level languages since they are closest to computer hardware.

High-level Programming Languages

- High-level programming languages create computer programs using instructions that much easier to understand than machine or assembly language instructions.

- Programs written in a high-level language must be translated into a low level language using a program called compiler.

- **A compiler** is a program that translates programming code written in a high-level language into a low-level format.

- Once a program is written in a high-level language, it must also be translated into the machine language of the computer on which it will be run. This translation can be accomplished in two ways:

  - ✓ When each statement in a high-level source program is translated individually and executed immediately upon translation, the programming language used is called an ***interpreted language***, and the program doing the translation is called an ***interpreter***.

  - ✓ When all of the statements in a high-level source program are translated as a complete unit before any one statement is executed, the programming language used is called a ***compiled language***. In this case, the program doing the translation is called a ***compiler***.

- Because of the difficulty of working with low-level languages, high-level languages were developed to make it easier to write computer programs.

- High-level languages allow programmers to write instructions that look like every English sentences and commonly used mathematical notations.

- So. High level programming languages create computer programs using instructions that are much easier to understand than machine or assembly language code because you can use words that more clearly describe the task being performed.

- Each line in a high-level language program is called a **statement**.

  Example:

  Result = (First + Second)*Third.

- **Examples:** high-level languages include FORTRAN, COBOL, BASIC, PASCAL, C, C++ and JAVA.