COMPUTER SYSTEMS[*]

Huong Nguyen

This work is produced by The Connexions Project and licensed under the
Creative Commons Attribution License [†]

A computer is an electronic device that performs calculations on data, presenting the results to humans or other computers in a variety of (hopefully useful) ways. The computer system includes not only the hardware, but also software that are necessary to make the computer function.

Computer hardware is the physical part of a computer, including the digital circuitry, as distinguished from the computer software that executes within the hardware.

Computer software is a general term used to describe a collection of computer programs, procedures and documentation that perform some task on a computer system

1 Computer Organization

1.1 General Model of a Computer

A computer performs basically five major operations or functions irrespective of their size and make.

1. Input: This is the process of entering data and programs in to the computer system. You should know that computer is an electronic machine like any other machine which takes as inputs raw data and performs some processing giving out processed data. Therefore, the input unit takes data from us to the computer in an organized manner for processing.

2. Storage: The process of saving data and instructions permanently is known as storage. Data has to be fed into the system before the actual processing starts. It is because the processing speed of Central Processing Unit (CPU) is so fast that the data has to be provided to CPU with the same speed. Therefore the data is first stored in the storage unit for faster access and processing. This storage unit or the primary storage of the computer system is designed to do the above functionality. It provides space for storing data and instructions.

The storage unit performs the following major functions:
- All data and instructions are stored here before and after processing.
- Intermediate results of processing are also stored here.

3. Processing: The task of performing operations like arithmetic and logical operations is called processing. The Central Processing Unit (CPU) takes data and instructions from the storage unit and makes all sorts of calculations based on the instructions given and the type of data provided. It is then sent back to the storage unit.

4. Output: This is the process of producing results from the data for getting useful information. Similarly the output produced by the computer after processing must also be kept somewhere inside the computer before being given to you in human readable form. Again the output is also stored inside the computer for further processing.

---

[*]Version 1.1: Jul 7, 2009 6:05 am -0500
[†]http://creativecommons.org/licenses/by/3.0/

5. Control: The manner how instructions are executed and the above operations are performed. Controlling of all operations like input, processing and output are performed by control unit. It takes care of step by step processing of all operations in side the computer.

In order to carry out the operations mentioned above, the computer allocates the task between its various functional units. The computer system is divided into several units for its operation.

- CPU (central processing unit) : The place where decisions are made, computations are performed, and input/output requests are delegated
- Memory: stores information being processed by the CPU
- Input devices : allows people to supply information to computers
- Output devices : allows people to receive information from computers
- Buses : a bus is a subsystem that transfers data or power between computer components inside a computer.
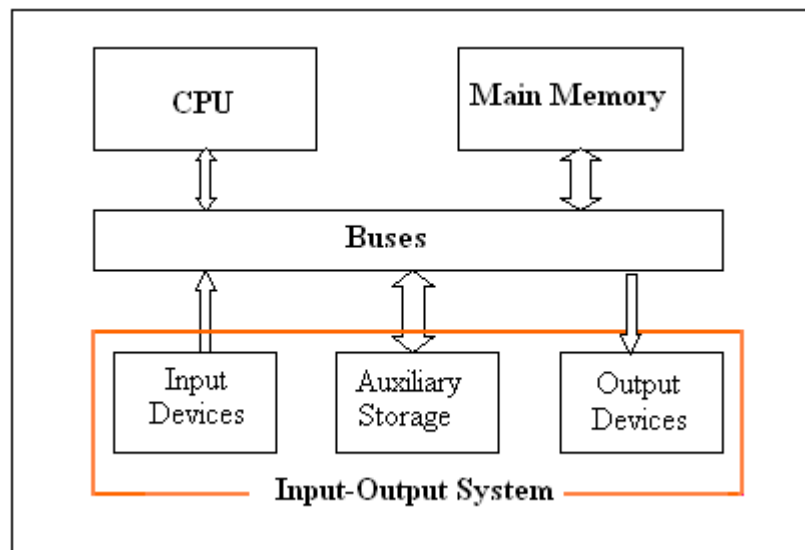

Figure 1: General model of a computer

The basic function of a computer is program execution. When a program is running the executable binary file is copied from the disk drive into memory. The process of program execution is the retrieval of instructions and data from memory, and the execution of the various operations.The cycles with complex instruction sets typically utilize the following stages :

Fetch the instruction from main memory

The CPU presents the value of the program counter (PC) on the address bus. The CPU then fetches the instruction from main memory via the data bus into the Memory Data Register (MDR). The value from the MDR is then placed into the Current Instruction Register (CIR), a circuit that holds the instruction so that it can be decoded and executed.

Decode the instruction

The instruction decoder interprets and implements the instruction.

Fetch data from main memory

Read the effective address from main memory if the instruction has an indirect address. Fetch required data from main memory to be processed and placed into registers.

Execute the instruction

From the instruction register, the data forming the instruction is decoded by the control unit. It then passes the decoded information as a sequence of control signals to the relevant function units of the CPU to perform the actions required by the instruction such as reading values from registers, passing them to the Arithmetic logic unit (ALU) to calculate the result and writing the result back to a register. A condition signal is sent back to the control unit by the ALU if it is involved.

Store results

The result generated by the operation is stored in the main memory, or sent to an output device. Based on the condition feedback from the ALU, the PC is either incremented to address the next instruction or updated to a different address where the next instruction will be fetched. The cycle is then repeated.

1.2 The Central Processing Unit (CPU)

You may call CPU as the brain of any computer system. It is the brain that takes all major decisions, makes all sorts of calculations and directs different parts of the computer functions by activating and controlling the operations.

CPU has four key parts

- Control Unit
- Arithmetic & Logic Unit
- Registers
- Clock

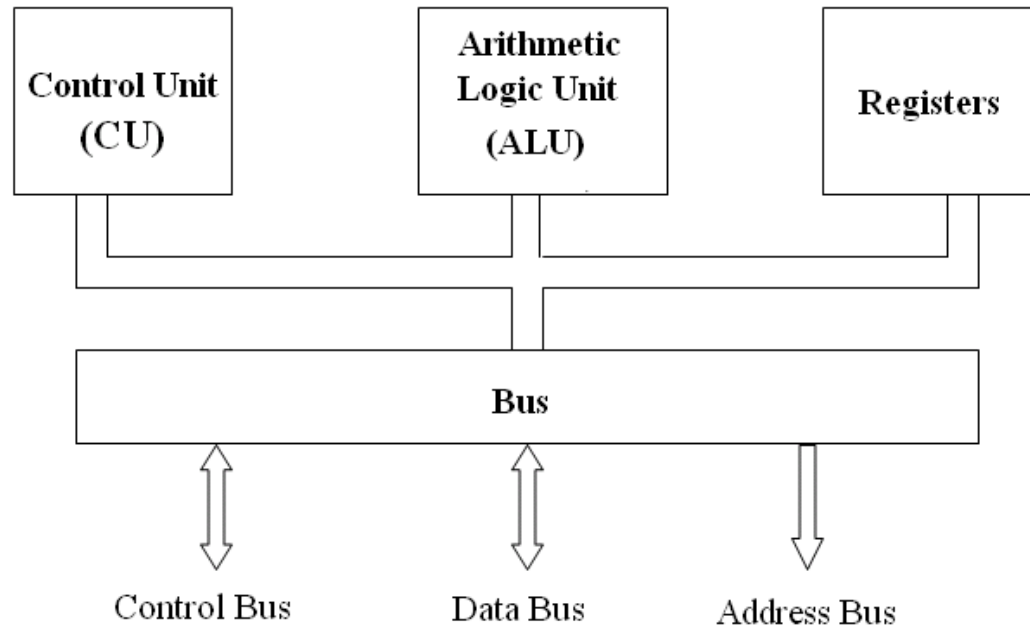And, of course, wires that connect everything together.

Figure 2: Basic Model of the Central Processing Unit (CPU)

Arithmetic Logic Units (ALU)

The ALU, as its name implies, is that portion of the CPU hardware which performs the arithmetic and logical operations on the binary data .The ALU contains an Adder which is capable of combining the contents of two registers in accordance with the logic of binary arithmetic.

Control Unit

The control unit, which extracts instructions from memory and decodes and executes them, calling on the ALU when necessary.

Registers

Registers are temporary storage units within the CPU. Some registers, such as the program counter and instruction register, have dedicated uses. Other registers, such as the accumulator, are for more general purpose use.

Clock

A circuit in a processor that generates a regular sequence of electronic pulses used to synchronize operations of the processor's components. The time between pulses is the cycle time and the number of pulses per second is the clock rate (or frequency).

The execution times of instructions on a computer are usually measured by a number of clock cycles rather than seconds. The higher clock rate, the quicker speed of instruction processing. The clock rate for a Pentium 4 processor is about 2.0, 2.2 GHz or higher

1.3 Memory

Memory refer to computer components, devices and recording media that retain digital data used for computing for some interval of time. Computer memory includes internal and external memory.

Internal memory

The internal memory is accessible by a processor without the use of the computer input-output channels.It usually includes several types of storage, such as main storage, cache memory, and special registers, all of which can be directly accessed by the processor.

Cache memory : A buffer, smaller and faster than main storage, used to hold a copy of instructions and data in main storage that are likely to be needed next by the processor and that have been obtained automatically from main storage.

Main memory (Main Storage) : addressable storage from which instructions and other data may be loaded directly into registers for subsequent execution or processing.

Storage capacity of the main memory is the total amount of stored information that the memory can hold. It is expressed as a quantity of bits or bytes. Each address identifies a word of storage. So the capacity of the main memory depends on the number of bits allowed to address. For instance, a computer allows also 32-bit memory addresses; a byte-addressable 32-bit computer can address 2^{32} = 4,294,967,296 bytes of memory, or 4 gigabytes (GB). The capacity of the main memory is 4 GB.

The main memory consists of ROM and RAM.

- Random Access Memory (RAM): The primary storage is referred to as random access memory (RAM) because it is possible to randomly select and use any location of the memory directly store and retrieve data. It takes same time to any address of the memory as the first address. It is also called read/write memory. The storage of data and instructions inside the primary storage is temporary. It disappears from RAM as soon as the power to the computer is switched off.
- Read Only Memory (ROM): There is another memory in computer, which is called Read Only Memory (ROM). Again it is the ICs inside the PC that form the ROM. The storage of program and data in the ROM is permanent. The ROM stores some standard processing programs supplied by the manufacturers to operate the personal computer. The ROM can only be read by the CPU but it cannot be changed. The basic input/output program is stored in the ROM that examines and initializes various equipment attached to the PC when the switch is made ON.

External Memory

The external memory holds information too large for storage in main memory. Information on external memory can only be accessed by the CPU if it is first transferred to main memory. External memory is slow and virtually unlimited in capacity. It retains information when the computer is switched off and is used to keep a permanent copy of programs and data.

Hard Disk: is made of magnetic material. Magnetic disks used in computer are made on the same principle. It rotates with very high speed inside the computer drive. Data is stored on both the surface of the disk. Magnetic disks are most popular for direct access storage device. Each disk consists of a number of invisible concentric circles called tracks. Information is recorded on tracks of a disk surface in the form of tiny magnetic spots. The presence of a magnetic spot represents one bit and its absence represents zero bit. The information stored in a disk can be read many times without affecting the stored data. So the reading operation is non-destructive. But if you want to write a new data, then the existing data is erased from the disk and new data is recorded. The capacity of a hard disk is possibly 20 GB, 30 GB, 40 GB, 60 GB or more.

Floppy Disk: It is similar to magnetic disk discussed above. They are 5.25 inch or 3.5 inch in diameter. They come in single or double density and recorded on one or both surface of the diskette. The capacity of a 5.25-inch floppy is 1.2 mega bytes whereas for 3.5 inch floppy it is 1.44 mega bytes. The floppy is a low cost device particularly suitable for personal computer system.

Optical Disk:With every new application and software (includes sounds, images and videos) there is greater demand for memory capacity. It is the necessity to store large volume of data that has led to the

development of optical disk storage medium. There are two commonly used categories of optical disks: CD with the approximate capacity of 700MB and DVD with the approximate capacity of 4.7GB

Memory Stick (Flash card, flash drive) a removable flash memory card format, with 128MB, 256 MB, 512 MB, 1 GB, 2 GB , 4 GB or more capacities
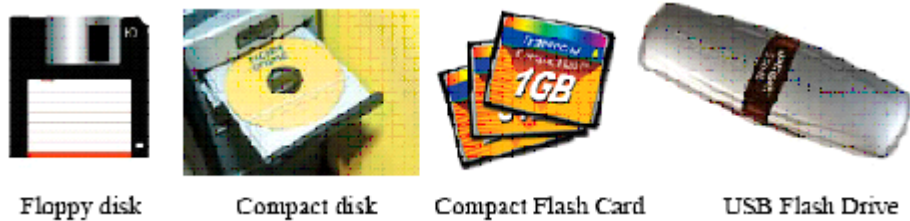
Floppy disk   Compact disk   Compact Flash Card   USB Flash Drive

Figure 3: Some types of auxiliary memory

1.4 Input-Output Devices

A computer is only useful when it is able to communicate with the external environment. When you work with the computer you feed your data and instructions through some devices to the computer. These devices are called Input devices. Similarly the computer after processing, gives output through other devices called output devices.

Common input and output devices are: Speakers, Mouse, Scanner, Printer,Joystick, CD-ROM, Keyboard, Microphone, DVD, Floppy drive, Hard drive, Magnetic tape, and Monitor. Some devices are capable of both input and output.
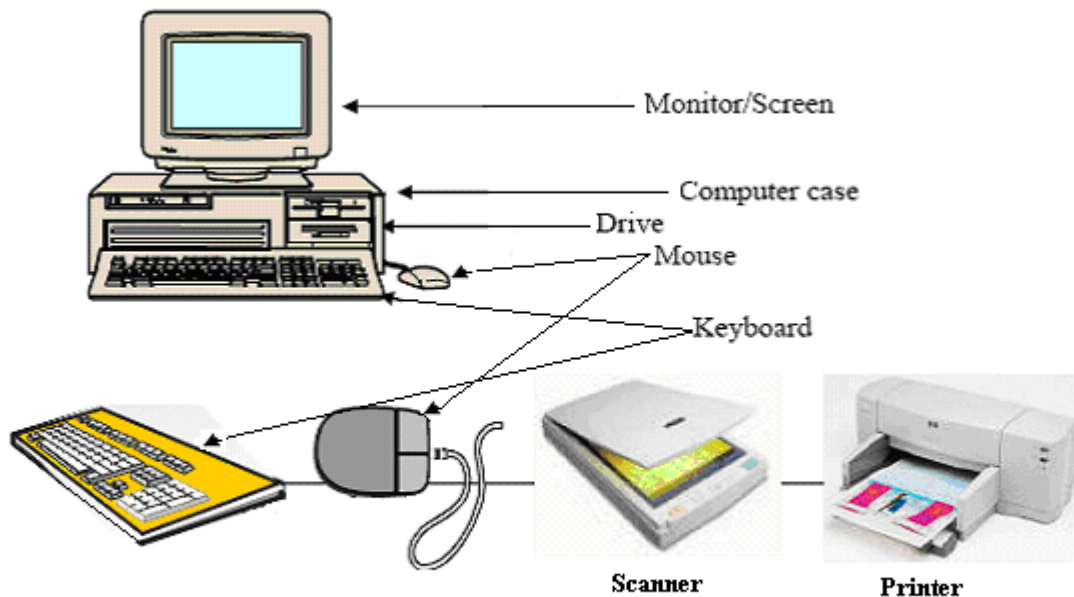
Figure 4: Typical input- output devices

Input Devices

Input devices are necessary to convert our information or data in to a form which can be understood by the computer. A good input device should provide timely, accurate and useful data to the main memory of the computer for processing followings are the most useful input devices.

Keyboard: - This is the standard input device attached to all computers. The layout of keyboard is just like the traditional typewriter. It also contains some extra command keys and function keys. It contains a total of 101 to 104 keys. You must press correct combination of keys to input data. The computer can recognize the electrical signals corresponding to the correct key combination and processing is done accordingly.

Mouse: - Mouse is an input device that is used with your personal computer. It rolls on a small ball and has two or three buttons on the top.When you roll the mouse across a flat surface the screen censors the mouse in the direction of mouse movement. The cursor moves very fast with mouse giving you more freedom to work in any direction. It is easier and faster to move through a mouse.

Scanner: The keyboard can input only text through keys provided in it. If we want to input a picture the keyboard cannot do that. Scanner is an optical device that can input any graphical matter and display it back.

Output Devices

Monitor: The most popular input/output device is the monitor. A Keyboard is used to input data and Monitor is used to display the input data and to receive massages from the computer. A monitor has its own box which is separated from the main computer system and is connected to the computer by cable. It can be color or monochrome. It is controlled by an output device called a graphics card. Displayable area

measured in dots per inch, dots are often referred to as pixels. Standard resolution is 640 by 480. Many cards support resolution of 1280 by 1024 or better. Number of colors supported varies from 16 to billions

Printer: It is an important output device which can be used to get a printed copy of the processed text or result on paper. There are different types of printers that are designed for different types of applications.

1.5 Buses

Bus is a subsystem that transfers data or power between computer components inside a computer or between computers. Bus can logically connect several peripherals over the same set of wires. Each bus defines its set of connectors to physically plug devices, cards or cables together. The buses are categorized depending on their tasks:

- The data bus transfers actual data.
- The address bus transfers information about where the data should go.
- The control bus carries signals that report the status of various devices.

2 Computer Software

2.1 Data and Algorithms

There are many steps involved in writing a computer program to solve a given problem. The steps go form problem formulation and specification, to design of the solution, to implementation, testing and documentation, and evaluation the solution.

Once we have a suitable mathematical model for our problem, we attempt to find a solution in term of that model. Our initial goal is to find a solution in the form of an algorithm. So what is an algorithm?

Algorithm is a finite sequence of instructions each of which has a clear meaning and can be performed with a finite amount of effort in a finite length of time.

How do you represent an algorithm? The most obvious representation: source code of a programming language. However, writing source code before you fully understand an algorithm often leads to difficult-to-find bugs. So, algorithms may be presented ...

1. In words

To present the algorithm in words we may describe the tasks step by step.

2.As a flowchart

A familiar technique for overcoming those bugs involves flowcharts.

A flowchart is a visual representation of an algorithm's control flow. That representation illustrates statements that need to execute, decisions that need to be made, logic flow (for iteration and other purposes), and terminals that indicate start and end points.

To present that visual representation, a flowchart uses various symbols, which Figure 3.5.shows.
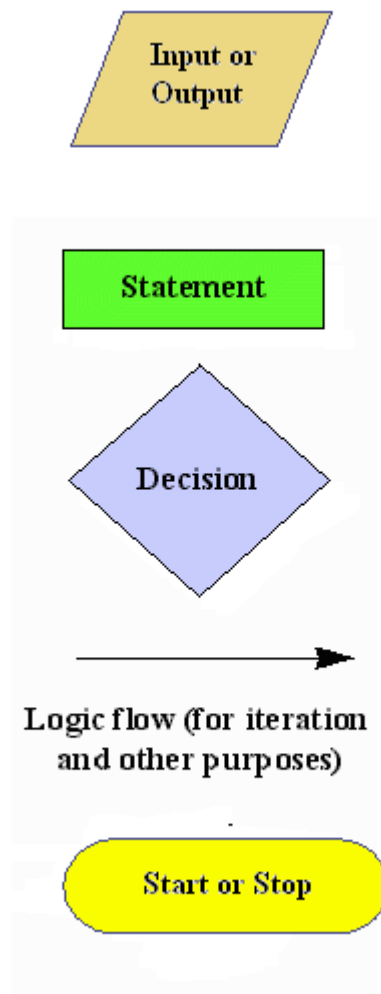
Figure 5: Flowchart symbol for statements, decisions, logic flows, etc.

This review was essential because we will be using these building blocks quite often today.

3. In pseudocode

Pseudocode (derived from pseudo and code) is a compact and informal high-level description of a computer programming algorithm that uses the structural conventions of programming languages, but omits detailed subroutines, variable declarations or language-specific syntax. The programming language is augmented with natural language descriptions of the details, where convenient, or with compact mathematical notation.

Example

Present the algorithm of converting an integer from decimal to binary

a. By words

Step 1: Let x is the decimal integer you want to convert and let k=1

Step 2 : Divide x by 2, saving the quotient as Q, and the remainder (in binary) as R_k

Step 3 : If Q is not zero, let X=Q, and go back to step 2. Otherwise go to step 4.
Step 4 : Assume step 1-3 were repeated n times. Arrange the remainders as string for digit $R_n R_{n-1}...R_3 R_2 R_1$.
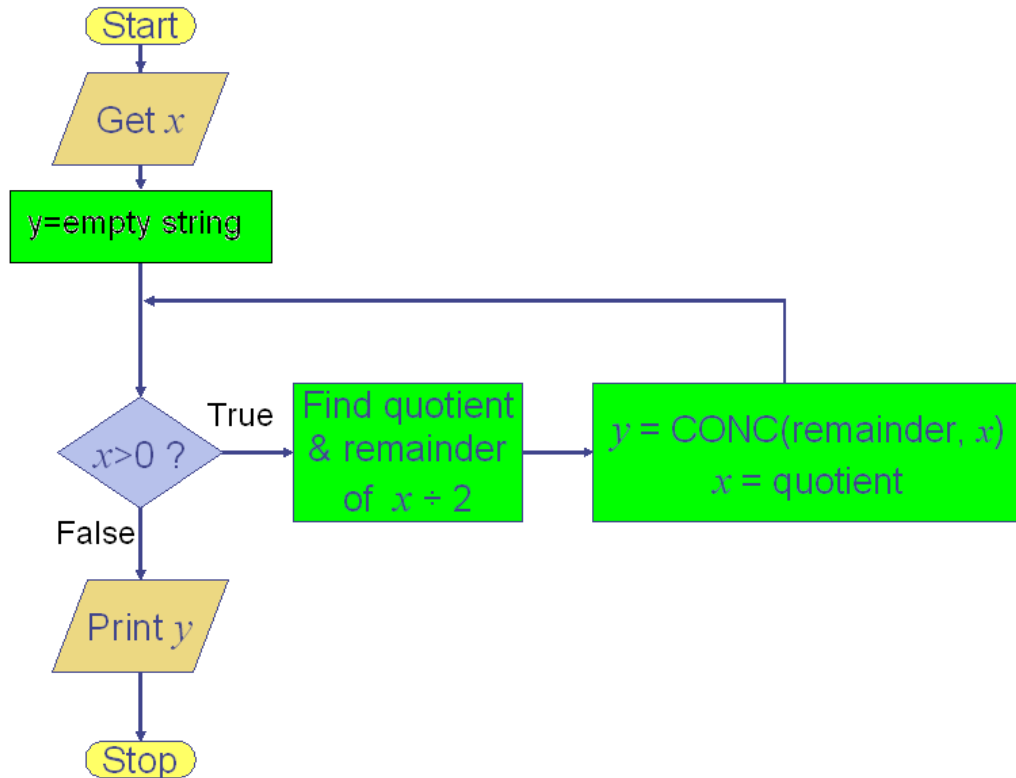b. As a flowchart

Figure 6: Flowchart of the algorithm of converting an integer from decimal to binary

c. By pseudocode

```
BEGIN
input x.
y=''''
remainder=0,
  while (x>0)
begin
  quotient=x/2
  remainder=x mod 2
  y=conc(remainder,y)
  x=quotient
end
print y
END.
```

Example

Bubble Sort

Bubble sort is a simple sorting algorithm. It works by repeatedly stepping through the list to be sorted, comparing two items at a time and swapping them if they are in the wrong order. The pass through the list is repeated until no swaps are needed, which indicates that the list is sorted.

5 1 4 2 8 - unsorted array

1 4 2 5 8 - after one pass

1 2 4 5 8 - sorted array

The algorithm gets its name from the way smaller elements "bubble" to the top (i.e. the beginning) of the list via the swaps.Because it only uses comparisons to operate on elements, it is a comparison sort. This is the easiest comparison sort to implement.

Here are the presentations of bubble sort algorithm

a. By words

Step 1: Get the length of the list : N and the list: list[1],list[2],…,list[N]

Step 2: $M \leftarrow N$.

Step 3: If $M < 2$ then print the list, stop.

Step 4: $M \leftarrow M - 1$, $i \leftarrow 0$.

Step 5: Increase i by 1

Step 6: If $i > M$ then go to step 3.

Step 7: If list[i] > list[i+1] swap list[i] and list[i+1]
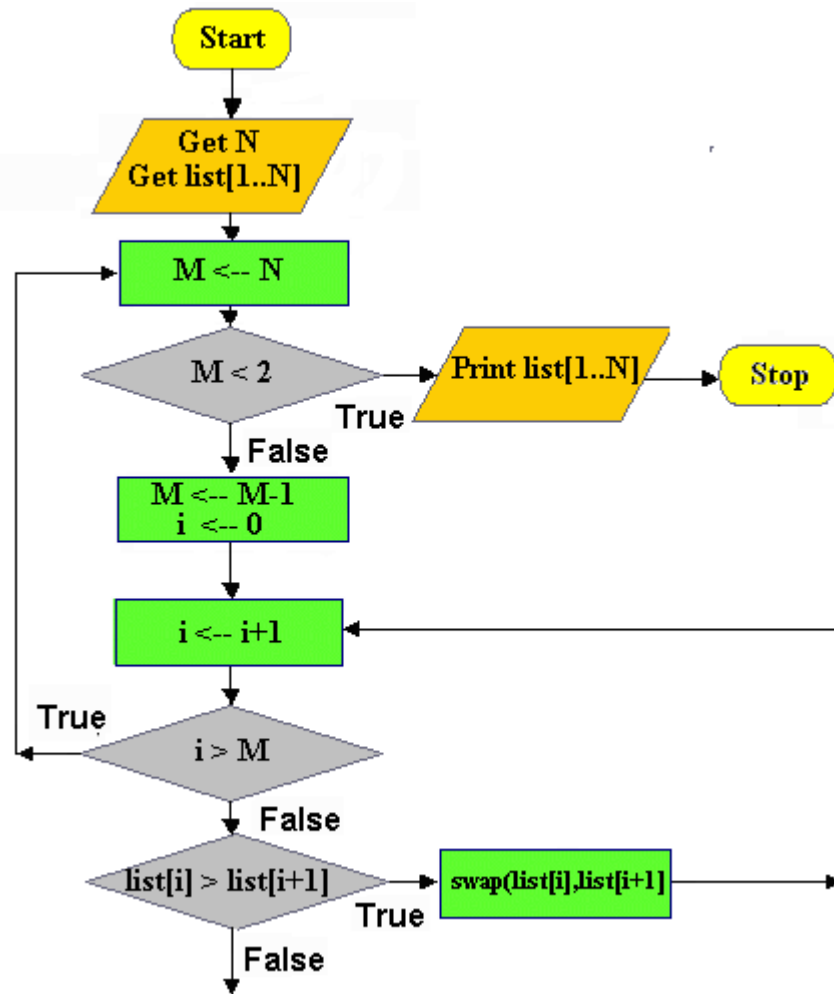
Step 8: Go to step 5.

b. As a flow chart

Figure 7: Flowchart of bubble sort algorithm

c. In pseudocode

A simple way to express bubble sort in pseudocode is as follows:

```
BEGIN  get length (list) and list's elements
  for each M in length(list) down to 2 do:
     for each i  in 1 to M-1 do:
        if list[ i] > list[ i+1 ] then
          swap( list[i+1],  list[ i ] )
        end if
     end for
  end for
end procedure
```

Comparing the three methods,especially pseudocode and flowchart we realized :

Pros and Cons of Flowcharts

In fact, flowcharts are not very useful.The process of writing an algorithm in the form of a flowchart is just too cumbersome, and then converting this graphical form into code is not straight forward

However, there is another kind of flowcharts – called Structured Flowcharts – that may be better suited for software developers.

The good thing about flowcharts is that their symbols are quite intuitive and almost universally understood. Their graphical nature makes the process of explaining an algorithm to one's peers quite straightforward.

Pros and Cons of Pseudocode

Pseudocode are quite suitable for software development as it is closer in form to real code.One can write the pseudocode, then use it as a starting point or outline for writing real code.

Many developers write the pseudocode first and then incrementally comment each line out while converting that line into real code.Pseudocode can be constructed quite quickly as compared with a flowchart.

Unlike flowcharts, no standard rules exist for writing pseudocode

To design an algorithm, the following characteristics are very

- Exactness
- Effectiveness
- Guaranteed termination
- Generality

The concept of structured programming says that any programming logic problem can be solved using an appropriate combination of only three programming structures,

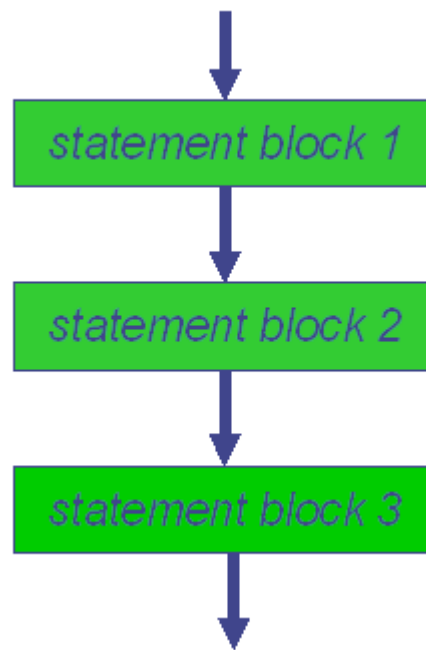**1.Sequence**: a sequence of instructions that are executed in the precise order they are written in

Figure 8

2. **Conditional** : Select between alternate courses of action depending upon the evaluation of a condition
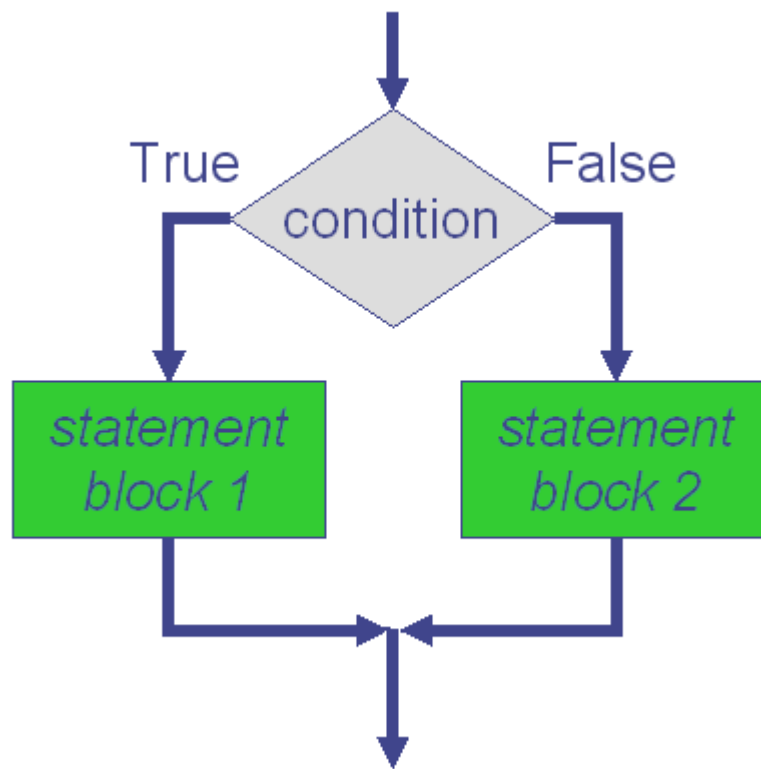
Figure 9

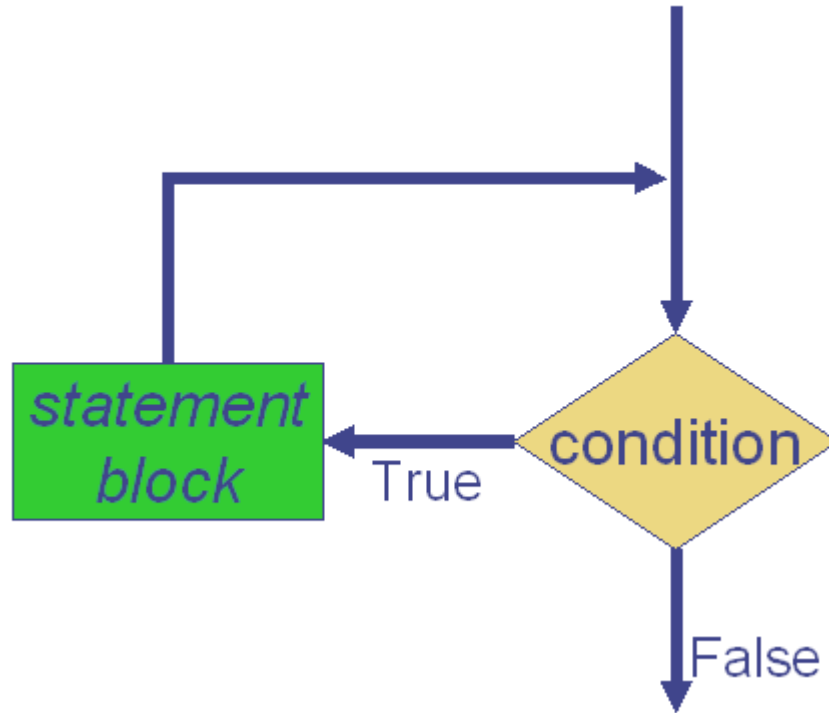**Loops**: Loop through a set of statements as long as a condition is true

Figure 10

2.2 Programs and Programming Languages

Programs

A computer program is an algorithm written for a computer in a special programming language.

Programming languages

A programming language is an artificial language that can be used to control the behavior of a machine, particularly a computer. It is defined through the use of syntactic and semantic rules, to determine structure and meaning respectively.

Programming languages are used to facilitate communication about the task of organizing and manipulating information, and to express algorithms precisely.

There are large number of programming language in use. We can identify three type of programming languages : machine languages, assembly languages, high-level languages.

Machine Languages

Machine code or machine language is a system of instructions and data directly executed by a computer's central processing unit. Machine code is the lowest-level of abstraction for representing a computer program.Instructions are patterns of bits with different patterns corresponding to different commands to the machine. Machine code has several significant disadvantages : very difficult for a human to read and write, a program written on one computer cannot run on a different computer, so it cannot be used to write large program or program intended to run on different machines.

Assembly Languages

An assembly language is a low-level language for programming computers. It implements a symbolic representation of the numeric machine codes and other constants needed to program a particular CPU architecture.

This representation is usually defined by the hardware manufacturer, and is based on abbreviations (called mnemonics) that help the programmer remember individual instructions, registers, etc. An assembly language is thus specific to a certain physical or virtual computer architecture

A utility program called an assembler, is used to translate assembly language statements into the target computer's machine code.

Although assembly is more friendly than machine code, use of assembly offer several disadvantages, for instance, each type of computer has its own assembly language or programming assembly requires much time and effort.

Hence, assembly language is not use to write large programs. However, there are some computer application, such as in writing program that control peripherals, assembly is still a necessity.

High-level languages

A high-level programming language is a programming language that, may be more abstract, easier to use, or more portable across platforms.

Examples: Pascal, C, Visual Basic, SQL, ...

Such languages often abstract away CPU operations such as memory access models and management of scope.These languages have been implemented by translating to machine languages.

There are two types of translators

- **Compiler** is a program that translate source code from a high-level programming language to a lower level language (e.g., assembly language or machine language)

- **Interpreter** is a program that translates and executes source language statements one line at a time.

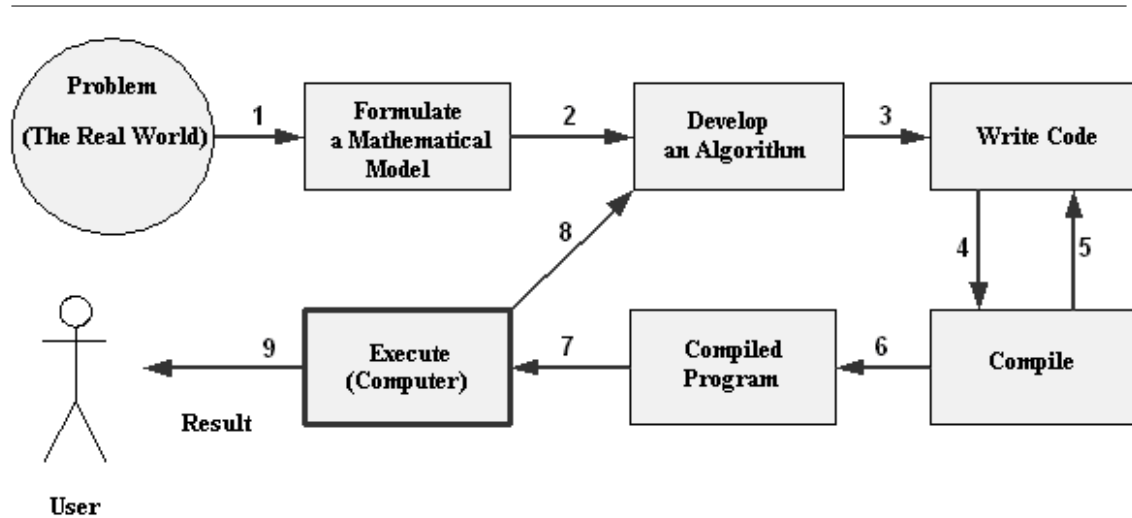Figure 11 below shows the process of solving problem with computers

Figure 11: Steps in software development

Domain Analysis

Often the first step in attempting to design a new piece of software, whether it be an addition to an existing software, a new application, a new subsystem or a whole new system, is, what is generally referred to as "Domain Analysis". The more knowledgeable they are about the domain already, the less the work required. Another objective of this work is to make the analysts who will later try to elicit and gather the requirements from the area experts or professionals, speak with them in the domain's own terminology and to better understand what is being said by these people. Otherwise they will not be taken seriously. So, this phase is an important prelude to extracting and gathering the requirements.

Software Elements Analysis

The most important task in creating a software product is extracting the requirements. Customers typically know what they want, but not what software should do, while incomplete, ambiguous or contradictory requirements are recognized by skilled and experienced software engineers. Frequently demonstrating live code may help reduce the risk that the requirements are incorrect.

Specification

Specification is the task of precisely describing the software to be written, possibly in a rigorous way. In practice, most successful specifications are written to understand and fine-tune applications that were already well-developed, although safety-critical software systems are often carefully specified prior to application development. Specifications are most important for external interfaces that must remain stable.

Software architecture

The architecture of a software system refers to an abstract representation of that system. Architecture is concerned with making sure the software system will meet the requirements of the product, as well as ensuring that future requirements can be addressed. The architecture step also addresses interfaces between the software system and other software products, as well as the underlying hardware or the host operating system.

Implementation (or coding)

Reducing a design to code may be the most obvious part of the software engineering job, but it is not necessarily the largest portion.

Testing

Testing of parts of software, especially where code by two different engineers must work together, falls to the software engineer.

Documentation

An important (and often overlooked) task is documenting the internal design of software for the purpose of future maintenance and enhancement. Documentation is most important for external interfaces.

2.3 Classification of Computer Software

The software is divided to System Software and Application Software with each having several sub levels.

System software is the low –level software required to manage computer resources and support the production or execution of application program.

Application software is software program that perform a specific function directly for the end user.

System Software includes

- Operating Systems software
- Network Software : network management software, server software, security and encryption software, etc.
- Database management software
- Development tools and programming language software: software testing tools and testing software, program development tools, programming languages software
- Etc.

Application Software includes

- General business productivity applications : software program that perform a specific function directly for the end user, examples include : office applications, word processors, spreadsheet, project management system ,etc.
- Home use applications : software used in the home for entertainment, reference or educational purposes, examples include games, home education etc.
- Cross-industry application software : software that is designed to perform and/or manage a specific business function or process that is not unique to a particular industry, examples include professional accounting software, human resources management, Geographic Information Systems (GIS) software, etc.
- Vertical market application software : software that perform a wide range of business functions for a specific industry such as manufacturing, retail, healthcare , engineering, restaurant, etc.
- Utilities software : a small program that performs a very specific task. Examples include : compression programs, antivirus, search engines, font, file viewers, voice recognition software, etc.