CHAPTER

1 Fundamentals of Computers

Welcome to the World of Computers and Programming

This book provides you the opportunity to realize the potential in you. You are not new to this world. Your brain is the best computer ever created and the programming logic is inside your brain. Just use your common sense while reading the book to realize the potential in you. The best definition of intelligent is one who uses common sense in every aspect of his life. Definitely, you have common sense. Just use it and become intelligent. Enjoy the reading.

Engineers build the Nation

- Pandit Jawaharlal Nehru – The First Prime minister of India

Science deals with natural things whereas engineering creates artificial things. Engineering is about problem solving and nature has solved many complex problems and is a source of inspiration for solving many engineering problems. Scientific knowledge enables the use of the systematic procedure for solving the problems. Scientific methods are employed for solving engineering problems. Experience makes complex problems appear as simple and efficiency and effectiveness are always desired in problem solving. Today's engineer has to solve complex problems and is a global engineer.

Globalization is aimed at reducing the gap between the Nations and enables people and markets to view the world from the same perspective. Though the main aim may be considered as enabling the international trade, there are many other goals which are also important. Universal recognition of the degrees offered by different nations is one of the main results of globalization. Different nations have different standards of education making the mobility of the students and intellectuals difficult. In simple terms, mutual recognition and equality of the degrees given by different nations is required. The Washington accord is a global agreement among the bodies responsible for accreditation of engineering programs in various nations. India is one of the permanent signatories.

Accreditation Board of Engineering and Technology (ABET) is the accreditation board of United States of America (USA). It defines Engineering as "the profession in which knowledge of the mathematical and natural sciences gained by study, experience, and practice is applied with judgment to develop ways to utilize economically the materials and forces of nature for the benefit of mankind".

ABET also defines Engineering Technology as "the part of the technological field that requires the application of scientific and engineering knowledge and methods combined with technical skills in support of engineering activities; it lies in the occupational spectrum between the craftsman and the engineer at the end of the spectrum closest to the engineer".

[1] Computer science and engineering, apart from being a professional engineering field, is also influencing all other engineering and non-engineering fields. Computer programming is an integral part of a computer science curriculum and is also one of the essential components of other engineering fields. Since programming knowledge is essential for understanding various subjects of the computer science program, it is studied at first year under-graduation level. Many students find programming difficult to understand. Failing to perform better in computer programming subject discourages the students and affects their performance in the rest of the subjects in computer science curriculum. Many teachers also feel it difficult, to teach programming to students because the dynamics of programming subject are different from other traditional subjects. Computer Science deals with non-tangible objects i.e., software, whereas other branches of engineering deal with tangible objects. The fact is, it is the non-tangible objects that control the tangible objects. The development of software involves solving problem computationally. Computational problem solving requires computational thinking.

Computational Thinking

A new way of thinking is necessary to solve problems using computers. Think yourself as a computer rather than as the user of the computer to solve the problems using a computer. This thinking is called as computational thinking. Computational thinking as stated by Prof. Jeannette Wing refers to "the set of concepts and strategies used by computer scientists to formulate and solve problems".

The other definitions of computational thinking are:

- Making the computer do what you want to do.
- Working of computers and humans together to solve computational problems.
- It is the ability to solve the computational problems using one's own thinking abilities, those of other people, and using the computer as a tool.

1.1 Problem Solving

Engineering is about problem solving. All academic disciplines of engineering involve solving specific problems related to that discipline. We learn problem solving through informal and formal education and through reflective practice.

[1] If an university is giving engineering and Technology degree then students are both an Engineer and Craftsman.

Problem solving is considered as the most complex of all intellectual functions. Goldstem & Levin, define problem solving as "higher order cognitive process that requires the modulation and control of more routine or fundamental skills"

A problem is confronted with when we are in a situation where the solution to the problem is not obvious. We are stuck up and the direction to be taken is unknown. Problem solving can be considered as moving from a given initial situation to the desired goal situation. Solution to the problem is the steps to be taken to move from initial state to goal state.

The four necessary conditions for solving the problem as defined by David Moursund are:

- An initial situation or state is clearly known.
- The goal state is defined.
- You have the essential resources that enable you to move from the given initial situation to the desired goal situation.
- You have some ownership.

Initial and goal state must be clearly defined. Without a clear definition of the goal state, we cannot decide whether we have solved the problem or not. The initial situation help us in understanding the problem and formulate the problem. We have to investigate the resources that are required for solving the problem. Without the resources, we cannot solve the problem. Though the availability of the resources doesn't guarantee that solution to the problem can be determined but it increases the probability of finding the solution to the problem. The ownership indicates whether you are ready to spend time and use your abilities to solve the problem.

The problems can be classified into two types. Trivial problems where the problem statement is obvious and we need not put any additional effort in formulating the problem. Majority of the problems which we solve are of reasonable complexity and requires efforts from us to formulate the problem. Formulating the problem requires critical analysis of the problem, interaction with the customer to understand the problem from his perspective. After this, a problem has to be formulated which is clear and there is no scope for ambiguity.

Problem solving is not only a challenging activity, but it gives satisfaction to the problem solver and improves his confidence level. The problem solved need not be a complex one. There is an element of discovery in every problem that is solved. Ofcourse the level of satisfaction depends on the complexity of the problem that was solved, but not all can solve complex problems, at least not at the beginning.

Many students fail to write the programs on their own. It is not because they are not clear about the syntax and semantics of the programming language. They fail because they don't know problem solving aspects of the discipline. In computer science, the problem solving aspect is often neglected. Student directly tries to write the programs. They are oriented towards becoming typists rather than programmers or

designers. As engineers, they should first solve the problems from the computational perspective.

Polya's Problem Solving Steps

Engineering is about problem solving. George Polya is the mathematician who introduced four principles of problem solving which are valid even today. The four principles of Polya for problem solving are:

1. Understand the problem
2. Devise a plan
3. Carry out the plan
4. Loopback

Before solving any problem one has to understand the problem perfectly. All the words in the problem statement are to be understood thoroughly and there should be no scope for ambiguity. Understanding the problem perfectly enables the problem solver to decide whether all the information required to solve the problem is available in the problem statement or not. Misinterpretation of the words in the problem statement leads to finding a successful solution to a wrong problem which is not acceptable.

A problem cannot be solved by trial and error. Systematic and scientific procedures have to be adopted to solve the complex problems. This requires identification of a suitable plan for solving the problem. Proper planning only results by experience. The user should think of the similar problems which are already solved by him. The same strategies used to solve the previous problems may be applied to the current problem. Care should be taken here because the same problem can be solved using different strategies. The previous strategy need not be the best strategy. It is better to think about the alternative strategies also and choose the best one.

After deciding the plan to be followed, next step is to systematically apply the plan to solve the problem. Usually, it leads to a solution. Patience is required in solving any complex problem and one has to persist with the plan. Ofcourse it has some limitations. Sometimes the plan chosen need not lead to a solution. If the plan doesn't work out one should not hesitate to select another plan. Loop back principle is interesting. It allows us to know the plan that has worked and the plan that doesn't work. It acts as input to decide the plan for the future problems.

1.2 Fundamental Techniques to Solve the Problem

There are five fundamental techniques which can be used to solve the problem:

1. Divide-and-Conquer
2. Greedy method
3. Dynamic Programming
4. Backtracking
5. Branch-and-Bound

Divide and Conquer is the age-old technique used by humans for solving any problem. This was the technique used by kings to conquer other dynasties. This is how the British conquered INDIA (BHARAT). In divide and conquer technique the problem to be solved is divided into subproblems. The process continues until we get subproblems which can be solved directly without any division. The subproblems are solved and the solutions are integrated such that finally the solution to the problem in totality is obtained.

The greedy technique is generally used for solving an optimization problem. An Optimization problem is one in which, we are given a set of input values and objective function which has to be either maximized or minimized with respect to some constraints or conditions. It is called Greedy because it always tries to choose an optimum value (which maximizes or minimizes objective function). The choice is local, not global. Since the decision is based on local information only, it is not guaranteed to produce an optimal solution.

Dynamic programming which is similar to divide and conquer solves the subproblems but the results obtained are reused if possible in the solution to larger subproblems. The main difference is, dynamic programming is a bottom-up approach whereas divide and conquer is a top-down approach.

Backtracking as the name indicates checks whether the partial solution that is found will lead to a complete solution. If it is, the process is continued; otherwise, the search backtracks to the point where a choice is made. The other possible solution is studied for its suitability. If one of the search processes is successful then the solution is found, otherwise, there is no solution.

Branch-and-Bound is a general optimization technique similar to the backtracking method. But in this breadth first approach is used i.e., all the solutions are tried simultaneously. An attempt is made to reduce the search space. Though the complexity is high it leads to an optimal solution. It is used to solve the complex problems.

1.3 Information Revolution

The world has witnessed many great revolutions. The revolutions have brought many changes in the lives of the people. The **French Revolution** (1789–1799) destroyed the absolute monarchy and brought rights to the citizens of France. The British witnessed **Industrial Revolution** in 18^{th} and 19^{th} centuries and brought changes in various sectors like agriculture, manufacturing, mining and transport. The changes subsequently spread to other countries around the world. The **Russian Revolution** is the collective term for the series of revolutions in Russia in 1917, which destroyed the autocracy and led to the creation of the Soviet Union. All these involve bloodshed.

One revolution which has brought many changes to the lives of the people without any bloodshed is **Information Revolution.** Information is playing a central role in all the fields of Arts, Science, and Engineering. Infact, in INDIA there is an act called "Right to Information Act", whereby the citizens have the privilege of knowing the information of almost any kind.

Similar to the above revolutions, the evolution of computer has changed the way we live. Most of the activities of day-to-day life are performed with the help of a computer. Whether we use computer directly or not, many of our activities are done with the help of a computer. The computer has even penetrated our body. Experiments are on to replace parts of our body with computer controlled organs. Robots which perform many tasks which are normally done by human beings are controlled by computers.

Many people have contributed to the development of a computer, particularly mathematicians. It has all kinds of components mechanical, electrical, electronic, metallurgical, physics related etc. It has influenced all fields one can imagine.

The field of Computer Science and Engineering is a professional engineering field. It has influenced and continues to influence all fields of engineering, medicine, arts, and sciences, etc. It is difficult to imagine a field where computers are not used for one purpose or the other.

A computer is a machine that manipulates data according to a set of instructions. A computer is just hardware. It is the software which actually controls the working of a computer. Computers range from small pocket PC to Supercomputers. The capability of computers has changed enormously over the years. Yesterdays supercomputers are today's personal computers.

1.4 Applications of Computers

Computers are used for a wide range of applications. The applications for which computers are used are changing rapidly. Initially, computers were used for data processing, but now Games, E-commerce, etc., are the dominant applications which are driving the Computer market. The following are some of the applications of computers:

Number Crunching

This is the first application for which computers were used. Mathematical calculations can be performed within seconds using computers, which a human takes a considerable amount of time, even several years. Some of the fields like Weather forecasting involve huge processing of the data which requires a group of highly sophisticated computers.

Data processing

Many of the today's applications require storing of a large amount of data, accessing it quickly when required and processing it. A Database management system allows us to do these tasks. Examples are airline reservation system, stock market, etc. All these require computers interconnected using a computer network.

World Wide Web (WWW)

The WWW is a major application of the Internet. It is this application which made normal people get attracted towards the Internet. It allows users to access text, audio, and video easily. Technically, WWW is not same as the Internet. WWW is actually a Distributed system, which is a layer of abstraction over the Internet. It hides all the intrinsic problems associated with accessing data from the Internet.

E-Commerce

You can buy and sell anything using computers and Internet. You can browse and purchase products - books, music, toys, clothing, electronics, and much more at the websites of major retailers. You can also buy and sell used items through websites.

Email

Electronic Mail (E-mail) is a convenient way to communicate with others. Using e-mail, information can be transferred within seconds. E-mail can be accessed using cell phones also. One significant advantage of E-mail is that the recipient need not be online when the sender transmits a mail. All the emails transmitted go to the server, where they are stored for the recipients to retrieve later. E-mail transfer is not restricted to transfer of text. Pictures, audio, and video can be transferred as attachments.

Instant Messaging

Instant messaging is like having a real-time conversation with another person or a group of people. More commonly used term for it is chatting. When you type and send an instant message, the message is immediately visible to the recipient. Unlike email, sender and receiver have to be online (connected to the Internet) and in front of their computers at the same time.

Entertainment

This is the major application for which computers are being used. Image, audio, and video captured using a digital camera, can be easily transferred to a computer and played. You can share images, audio, and video by posting them on a website. You can also listen to music on your computer, either by importing (transferring to your computer) music from audio CDs or by purchasing songs from a music website or tune in to one of the thousands of radio stations that broadcast over the Internet. You can watch movies online. It is expected that by 2020 80% of the Internet traffic will be

based on video streaming. But unfortunately, computers are affecting the Entertainment market in the form of piracy.

Games

Thousands of computer games in every conceivable category are available to entertain you. It is the application which is generating a lot of revenues to the Internet market. The games can be downloaded to your computer to play or games can also be played online.

Health

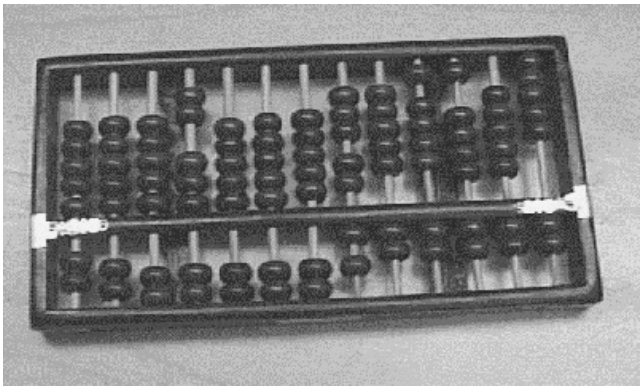
Many of the medical equipment used today are controlled by computers. A person's blood pressure, heartbeat, etc., can be continuously monitored using computers. Many long duration surgeries are being performed with the help of computers.

Control system

Many real-life scenarios cannot be controlled by humans manually. There are environments where it is difficult for humans to work. Examples are air traffic control, furnace temperature control, assembly line control in industry, etc. All these require sophisticated computers.

1.5 Evolution of Computers

A computer is basically a computing device. The concept of computing started even before the advent of computers. Fingers and pebbles were used for counting. The child starts counting using fingers. But the device which has revolutionized counting process and which is popular even today is **Abacus**. The Babylonians designed it in 500 B.C. There is an argument that calculations can be made faster using an abacus than using a calculator. An Abacus consists of just a few rods and free moving beads. But the limitation of Abacus is, it can be used only for computations on numerical values.



Arabs invented the decimal number system (zero was invented by Indians). Decimal number system consists of digits 0-9, and a number is formed by combinations of digits. Decimal number system made calculations on numbers simpler.

In 1837, Charles Babbage called "Father of Computing" started his work on the design of a general purpose computer called the Analytical Engine. Though the working Analytical Engine could not be materialized it served as a model to the next generation of computers. It was huge in size when compared to today's computers. The steam engine was used to supply power and accepted both data and programs. It has an Arithmetic logic unit to perform computations, storage to store programs, etc. Historically it was called the "First programmable computer".

The electrical charge was the invention of physics. With this invention, electrical charge was used to represent data. The presence of electrical charge represents a 1 and its absence a 0, which is a binary number system. The binary number system is preferred for representing data in a computer because of high instability of electronic devices.

Mathematician George Boole introduced Boolean algebra which eased the computation process of a computer. Claude Shannon showed to the world, use of Boolean algebra to manage switches inside a computer.

Theoretical Computer science was introduced to the world by Alan Turing, who is called the Father of modern computing. The backbone of the department of computer science and engineering in many of the premier institutes of India, like IIT's is theoretical computer science. One of the main contributions of Turing is Turing machine, a purely theoretical computer which can be used for doing many computations.

Military contributed a lot to the development of different technologies. Because of the huge fund that is allocated to military requirements, many technologies are the result of the funding from the military. The ENIAC (Electronic Numerical Integrator and Computer) is one of the innovations due to the military. It was designed for United States army to calculate the trajectory of artilleries to hit enemy targets. The ENIAC was the first fully programmable machine capable of solving almost any mathematical problem and which is huge in size containing many vacuum tubes, switches, etc. The vacuum tubes were of the size of the electrical bulb and were used to represent data using electrical charge. The term 'bug' was coined at that time because of the problems caused by moths which were attracted by heat and light of ENIAC which lead to short-circuiting. The programming on ENIAC was a complex task.

The mathematician John von Neumann introduced the concept of stored program computer. Stored program computer is capable of storing both data and programs in computer memory. He introduced Von Neumann architecture which is a model for a

modern computer consisting of Central processing unit (CPU), Arithmetic logic unit, Control unit, Memory, and Input/Output unit.

A very significant invention which earned a Noble prize for its inventors is Transistor. The persons were William Shockley, John Bardeen, and Walter Brattain. The transistor is a semiconductor device capable of switching electronic signals. In modern computers, these are usually part of ICs. Because of the abundance of silicon, it was used to make a transistor.

Robert Noyce, along with Gordon Moore formed Integrated Electronics, now known as Intel a famous microprocessor manufacturer. The CPU reduced to the size of an IC is called a microprocessor. The microprocessor is the heart of any modern computer. Infact microprocessor is not restricted to a computer, but it is part of many other electronic devices including toys. Gordon Moore speculation that "the number of transistors on a microprocessor would double every two years" is a famous one which remained true for many years.

The first commercial personal computer, the MITS Altair, was fitted with an Intel 8800 processor. Advanced Micro Devices (AMD) reverse engineered the Intel 8800 processor and started the long-running Intel-AMD rivalry. MITS Altair was not successful as expected. The Apple I the first fully assembled computer was developed around the same time by Steve Wozniak and Steve Jobs. The two subsequently founded Apple Computer in Jobs' family garage.

1.6 Generations of Computers

The first generation of computers (1950-1958) were based on vacuum tubes. The computers were bulky and expensive. They used a lot of electricity and generated a lot of heat. Magnetic drums were used as memory. Input was given using punched cards and output was sent to the printer. The programming was done using machine language. No support for high-level programming languages was provided. Examples are ENIAC, UNIVAC-I, UNIVAC-II, IBM 702 and 650.

Transistor was the main element of the second generation (1956-1963) of computers. Size and cost were reduced. Also, electricity consumption was drastically reduced. Magnetic tape was used as external storage device. Input was given using punched cards and output was sent to a printer. Assembly language was the language of choice for programming. High-level languages started arriving. Examples are IBM 7000, UNIVAC-III, and Honeywell 400.

The third computer generation (1964-1971), was marked by the arrival of the Integrated Circuits (ICs). There was a drastic increase in speed and efficiency of computers. People started using computers. Operating systems were designed for

computers and keyboard and monitor were the input/output devices. Examples are IBM 360 and 370, UNIVAC 1108, and DEC PDP-8.

The fourth-generation computers (1971 onwards) brought many changes. Microprocessors were introduced to reduce the size of a computer to size of a palm. In 1980's companies like Apple and IBM introduced personal computers. Graphical user interfaces were supported and the pointer devices like mouse were developed. With the advent of the Internet, computers were connected using a computer network. All the modern computers come from this generation of computers.

Fifth generation computing devices can be considered as next-generation computers. Many innovative technologies are being tried. Use of quantum computing and Nanotechnology are the prominent technologies being experimented. Also, Artificial intelligence and Voice recognition are to be supported by next generation computers.

1.7 Computer and its Components

The diagram of Fig. 1.1 is a model of a computer. This is basically called the Von Neumann architecture. Only main parts of a computer are shown. In the following text, computer parts are discussed from a practical point of view.

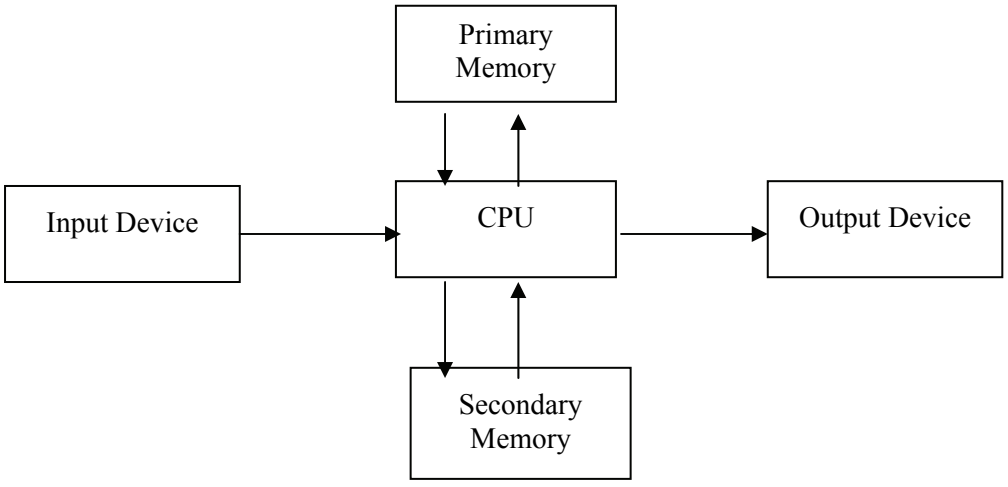
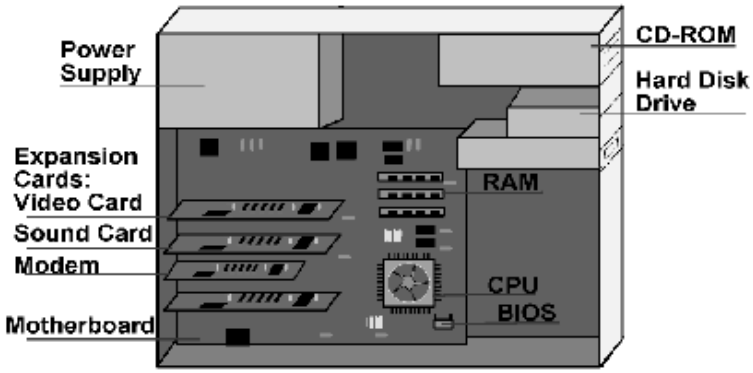


Fig. 1.1 Block Diagram of a Computer

Parts of a Computer

A side view of a computer showing a majority of the parts of a computer is shown below. The input/output devices are not shown in the diagram.



Case or Chassis

Case or Chassis is the enclosure for all the components of a computer. It is a framework to which all the components of a computer are attached directly or indirectly. Motherboard is securely bolted to the chassis and also Hard disk and CD drive. Components like expansion cards are fixed to the motherboard. Three types of chassis are available in the market: baby, desktop, and tower. The baby case is a small desktop case that can accommodate a minimum number of components of a computer. This case is generally used for a network terminal which depends on the server for most of its capabilities. Desktop cases come in a variety of shapes and sizes but can accommodate a reasonable number of components of a computer. The tower case is generally capable of holding the maximum number of components possible. The difference in the cases is mainly in the number of bays that can be attached to it. The bays are capable of holding hard disk and CD drives.

Central Processing Unit (CPU): The most important part of a computer is CPU. Infact computer is named based on the type of the CPU that is present inside the computer. For example, Pentium IV computer means the CPU is Pentium IV. It is called a central processing unit because there are other processors inside the computer which perform a specific task such as a numerical processor which performs mathematical calculations. When the central processing unit is reduced to the size of a small integrated circuit, it is called as a microprocessor or simply processor. Processors come in different varieties. Earlier computer systems used slot type processor and present computer systems use socket type processor. The two main components of CPU are Arithmetic logic unit and control unit. The arithmetic logic unit performs various arithmetic and logic operations. Control unit coordinates the overall functioning of the computer including input and output.

The microprocessors have changed a lot since their inception. Now a typical computer consists of more than 100 microprocessors. A typical computer also contains more than one CPU. Each CPU has a number of processing elements (arithmetic logic units) called cores.

Motherboard: This is printed circuit board (PCB) to which all the remaining components are connected. Some of the components are soldered into the motherboard, some are inserted into slots present on the motherboard and some are connected by a bus. The major functionality of a computer system is provided by motherboard.

The motherboard is the soul of a computer. The dimension and mounting points for the motherboard are typically defined by the form factor. The form factors available in the market are Baby, AT, and ATX. Different features are provided by ICs on the motherboard. Motherboards also come with expansion slots into which cards can be inserted which provides some functionality. Motherboard consists of CPU, chipset, busses, RAM, Soundcard, BIOS chip, Onboard I/O chip, Connectors, Battery, etc.

The chips on the motherboard have evolved to a stage where major functionality is provided just by a couple of related chips called chipsets. Majority of the external devices are attached to the motherboard through ports. Some of the ports are serial ports for devices like mouse and external modem, parallel ports for devices like printers, a special keyboard and mouse port, USB (universal serial bus) port through which variety of devices can be attached to the computer.

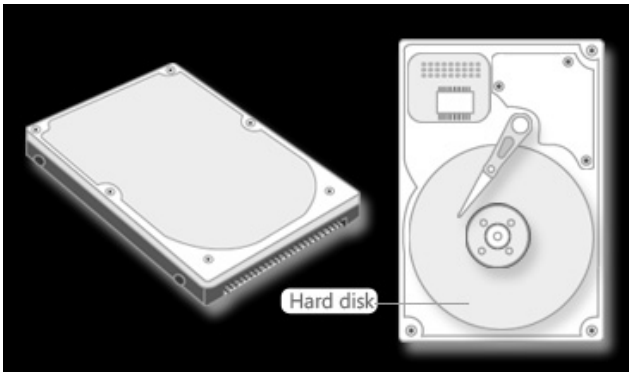
Power supply: This provides the required power for the operation of different components of a computer. The power supply used by the computer is called switched mode power supply (SMPS). Different components of a computer require different voltage levels. The components of a computer require DC (Direct Current). The SMPS converts AC (Alternating Current) to DC current and also distributes the power to different components of a computer based on their requirement. More the number of components inside a computer, more the capacity of SMPS.

Memory: Every computer has memory to store data and programs. Memory can be classified into different types based on the storage capacity, access speed and closeness to the CPU. Usually, the memory which is close to the processor or inside the processor is very fast but limited in capacity. The memory which is directly accessible by CPU is called Primary memory. The memory whose contents are to be brought into the primary memory before processing is called secondary memory. The following is the list of different types of memory used by a computer:

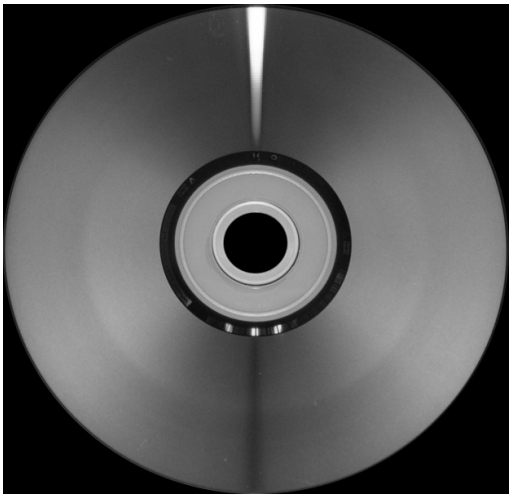
- **Random-access memory (RAM):** RAM is the primary memory. It is called as random access memory because any storage location can be accessed directly (randomly). RAM is used to temporarily store data because it is volatile and its contents are lost when the power goes off. To process data and to execute programs they have to be brought into primary memory. The primary memory can be imagined to be a collection of cells. Each cell has a memory address and each cell is capable of storing a byte of information. A single byte is 8 bits. The memory can be viewed as a mesh with rows and columns. The address value increases from left to right and top to bottom.

- **Read-only memory (ROM):** It is also Random access memory but it is read-only. Since its contents are not lost when the power goes off, it is a permanent memory storage used by the computer for important data that doesn't change. It is programmed during the manufacturing time and its contents generally are not modified. Special equipment is required to modify the contents of ROM.
- **Basic input/output system (BIOS):** A type of ROM that is used by the computer to establish basic communication when the computer is first powered on. It contains configuration information about the computer system.
- **Caching:** Cache is the fastest memory that is used by the computer system. CPU cache is used by the central processing unit of a computer to reduce the average time to access memory. The cache is a smaller, faster memory which stores copies of the data from the most frequently used main memory locations. There are three levels of cache L1, L2, and L3. It is difficult to clearly distinguish between these three levels of cache. The cache can be distinguished based on whether it is integrated with the processor or inside the processor chip or outside the processor chip. At present L1 cache is directly stored on the special section of the processor chip and L2 cache though is not part of the chip, but is on the same chip as that of the processor. L3 cache is a separate chip placed on the motherboard between the CPU and main memory.
- **Virtual memory:** RAM storage capacity is limited and the Hard disk has large storage capacity. Due to this a part of the space on the hard disk is used to temporarily store data. This data can be swapped into RAM when required and swapped out of RAM when is not required. The main advantage of virtual memory is, it allows execution of programs that are larger than the primary memory available. Remember it is not separate memory but is part of the Hard disk.
- **Flash memory:** It is a solid-state storage device which is nonvolatile and uses integrated circuits instead of magnetic or optical media. It is a technology that is primarily used in memory cards and USB flash drives for general storage and transfer of data between computers and other digital products. It is a specific type of EEPROM (Electrically Erasable Programmable Read-Only Memory) that is erased and programmed in large blocks. Because it requires erasing blocks of data before it can be written to, it has lower write performance. Flash memory gets its name because the microchip is organized so that a section of memory cells are erased in a single action or "flash.
- **Hard disk:** This is permanent storage and capable of storing a large amount of data. It is usually a stack of disks (not visible externally). Each such disk is divided into tracks and tracks further into sectors. Read/Write heads store and read the data from the hard disk. Hard disks record data by magnetizing Ferromagnetic material directionally. The small motor inside the hard disk enclosure moves the read/write heads. Hard disks are very sensitive and small dust particles can result in erasure of entire data and hard disk non-functional.

Some newer hard drives are flash-based with no moving parts. These drives are called solid state drives.

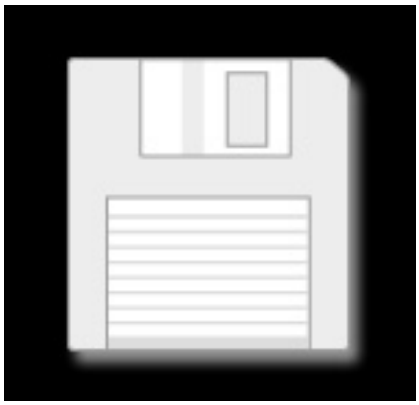


- **CD and DVD drives**
  Nearly all computers today come equipped with a CD (Compact Disk) drive or DVD (Digital Video Disk) drive, usually located on the front of the computer. It allows the users to insert the CD/DVD into the drive and perform reading/writing on the disk. CD drives use lasers to read data from a CD, and many CD drives can also write data onto CDs. CD drives which can read and write data are called CD writer.



DVD drives can do everything that CD drives can, plus read DVDs. Many DVD drives can record data onto blank DVDs. CD/DVD drives can be inserted into the computer and they also come in the form of external CD/DVD drives with USB (universal serial bus) interface. The disk drives appear similar to the Hard disk drive. DVD drives which can read and write data are called DVD writer.

- **Floppy disk drive**
  Floppy disk drives store information on floppy disks. Compared to CDs and DVDs, floppy disks can store only a small amount of data. They also retrieve data slowly and are more prone to damage. For these reasons, floppy disk drives are less popular than they used to be, although some computers still include them. Even though the outside of floppy is made of hard plastic, the inside of disk is made of a thin, flexible vinyl material. Floppy disks come in two types 5 ½ inches and 3 ½ inches. 3 ½ inch floppies are the ones which are used nowadays and have the storage capacity of 1.44MB. Floppy disk drives appear similar to CD/DVD drive but are smaller in size.



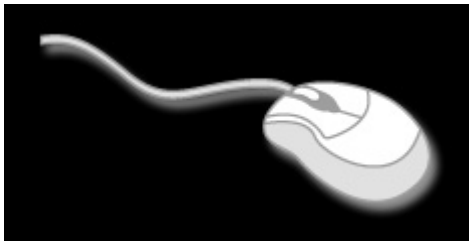
Note: One important point has to be remembered when estimating the amount of memory available. When storage capacity is represented in the form of KB, MB, and GB, K represents Kilo (2^{10}), M represents Mega (2^{20}), and G represents Giga (2^{30}). Often Kilo is wrongly interpreted as 10^3, Mega as 10^6 and Giga as 10^9. Hence when the memory capacity is 100KB, it is actually $100 \times 2^{10} = 102400$ bytes.

1.8 Basic I/O Devices

Mouse

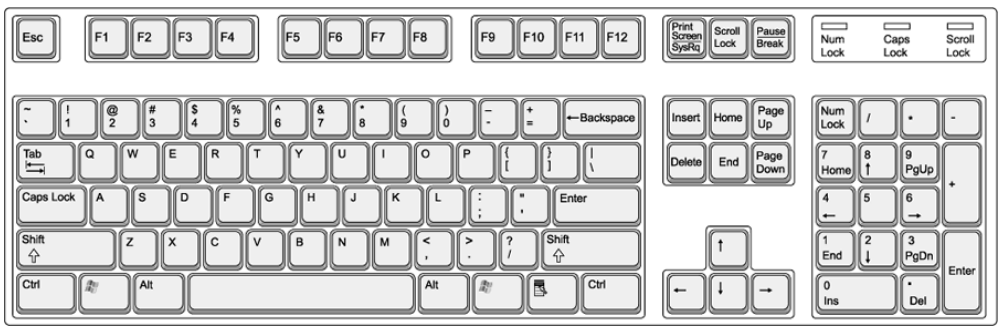
A mouse is a pointer device that can be used to point at the items on the screen and select items. The mouse is designed to support Graphical User Interface (GUI) environment. A mouse comes with two or three buttons and an optional wheel button. The left button is used for selecting a particular item on the screen by clicking the item. The right button is used to pop up a dialog box. The dialog box that is displayed depends on the present position of the mouse cursor on the screen. The wheel button is used to scroll through the contents which cannot be displayed on a single screen. Older mouse devices used to come with a ball at the bottom, and the movement of the ball changes the position of the mouse cursor. Current mouse devices are optical

mouse devices and they use the reflection of light to change the position of the mouse pointer. When the mouse is moved, a pointer on your screen moves in the same direction.



Keyboard

A keyboard is the fundamental input device. It is used to type text. A typical keyboard consists of 104 keys. Keyboards come in different configurations. The standard keyboard consists of all the generally used keys and multimedia keyboard consists of apart from the keys present in a standard keyboard, a sequence of keys related to multimedia like play, stop, and pause etc. Keys on the keyboard can be classified based on the purpose they serve. Alphabetic keys are used to type the text, Function keys perform different functions depending on when they are used, Numeric keypad keys located on the right side of most keyboards are for entering numbers quickly, Navigation keys such as the arrow keys for moving within a document or a webpage. Keyboards come in different configurations.



Monitor

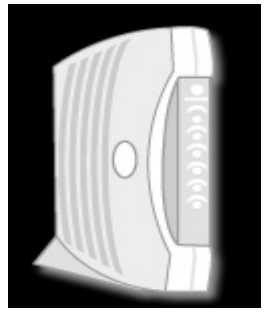
A monitor is a display device capable of displaying both text and graphics. Monitors can be broadly classified into two types: CRT (cathode ray tube) monitors and LCD (liquid crystal display) monitors. CRT monitors are bigger in size, consume more power but cost less. LCD monitors occupy less space, consume less power but cost more and have the problem of viewing angles i.e., the picture is not clear from certain angles. Recently introduced into the market are LED (light emitting diodes) monitors, which look similar to LCD monitors but which display video and graphics with good intensity. Usually, they adjust the brightness automatically depending on the light.

LED monitors can be called a type of LCD monitors because they use similar technology but differ only on backlighting. LCD monitors use fluorescent lamps whereas LED monitors use LED's for backlighting. LED monitors consume less power but relatively cost more than LCD monitors.

LCD monitor

CRT monitor

Modem



Modem stands for modulator and demodulator. A modem is a device that converts from analog form to digital form and digital form to analog form. It is used when data from a computer has to be transferred over analog transmission media such as telephone wires. Telephone wires carry analog data and the computer is digital, hence conversion is required. It is a communication device. Modems come in card form to insert them into a slot on the motherboard, and also in an external form whereby they can be attached to a computer using cable. Nowadays modems come inbuilt into the motherboard.

Ports: A port is an interface that allows a computer to communicate with peripheral equipment. All the peripheral equipment are connected through ports. Typical ports include COM ports, PS/2 ports, and USB ports. The ports can also be classified as serial ports and parallel ports.

Real-time clock: The different components of a computer need to be synchronized. This is achieved using a clock which is a vibrating crystal.

Complementary Metal-oxide Semiconductor (CMOS): The CMOS chip and CMOS battery allow a computer to store information even when the computer powers down. Usually, the BIOS password is stored in a memory which is powered by CMOS battery. As long as the battery is working, the password remains. When the battery is discharged/removed then some of the BIOS contents are lost.

Fans, Heatsinks, and Cooling systems: The components in a computer generate heat, particularly processor. As heat increases performance can suffer and it may lead to failure of the computer also. CPU is protected by a separate heat sink and a fan. Another fan protects the other components of a computer.

1.9 Computer Software

The two major elements of a computer are hardware and software. Hardware and software are generally integrated to such a level, that it is sometimes difficult to clearly distinguish the role played by software and hardware. It is software which controls the operation of hardware and through which variety of applications are accessed by the user of the computer.

Computer software provides a variety of functionality. The software can control the operation of the computer, provide users various applications, makes the computer convenient to use for a programmer, allows the programmer to add his own functionality by programming, entertain the user, allow the user to communicate, allows the user to prepare documents, presentations, etc.

Software can be broadly classified into two types:

- System software
- Application software

1.9.1 System Software

System software is computer software designed to operate the computer hardware and to provide a platform for running application software. The following are included in system software:

- Operating system
- Compiler/Interpreter/Assembler
- Loader/Linker
- Editor
- Debugger
- System Utilities

Operating system (OS) is a control program. It makes the computer usable. It allows the user to interact with the hardware through device drivers. Device drivers are software which understands the intricacies of the underlying hardware. Every device operates differently and requires special instructions to operate it correctly and achieve the expected functionality. For example, the keyboard has its own device driver and similarly mouse. Device drivers are written using languages which are close to hardware such as machine language and assembly language. The important goal in the design of device drivers is efficiency.

The operating system provides file related functionality by supporting different file systems. For example, Microsoft Windows support FAT, VFAT, NTFS file systems, and UNIX variants support 'ext' family of file systems like ext2, ext3, etc. A file system is a method of storing computer files and data. The operating system may classify the files into different types. For example, UNIX operating system doesn't

support any file types and any file is treated as a stream of bytes. Windows differentiates files based on file extension.

The operating system manages both primary and secondary memory. Programs that are to be executed are to be brought into the primary memory. Since primary memory is limited in size, memory allocation is an important issue. Secondary memory has to be organized such that a file from it can be quickly retrieved and stored. Swapping from primary memory to secondary memory and from secondary memory to primary memory is another important memory management function.

The program is passive entity occupying some space in secondary memory. A program ready to be executed is called a process. The operating system controls the execution of processes. Scheduling is the term used to allocate the CPU to a process to be executed. Also during execution, a process can be in different states. It is the operating system which maintains different states of a process. Also, the operating system should support protection mechanisms.

The ability to communicate with other systems is one of the expected functionalities of a computer. To support it networking software is required. Previously networking software was supported as add-on-modules but nowadays it is built into the operating system.

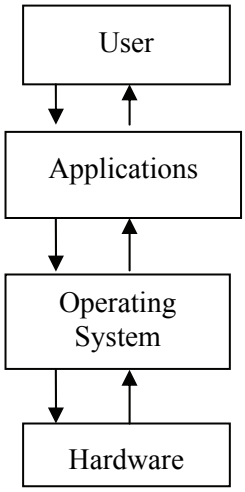


Fig. 1.2 Relation of OS with Hardware and Applications

A Computer Network is defined as an interconnected collection of computers. A computer network involves at least two computers. A computer can communicate with other computers using either wireless medium or wired medium.

Computer networks can be classified based on the distance between computers (other factors apart from distance can also be considered). If the distance is within 1 or 2 kilometers it is called as Local Area Network (LAN). LAN is usually limited to a

single organization. It doesn't spread across different organizations. If the distance is up to 10 km i.e., covers the entire city, then it is called as Metropolitan Area Network (MAN). If the distance is over 10 km and approximately up to 1000 km, the network is called Wide Area Network (WAN). Beyond it, it is an interconnection of different networks called the internet. The internet is a special kind of network based on TCP/IP technology.

A distributed system is a layer of abstraction created over a computer network. In a computer network, the presence of different computers is visible i.e., the users should know where the different resources or services are available and have to explicitly access the resources or services by logging into the system. In a distributed network, it is up to the system to find the suitable resources and services and make them available to the users.

There are different system software's which help the programmer in writing the application programs. Editors are required for writing the program and edit it. Usually, most of the programming languages come with integrated development environment which integrates editing, compiling and executing programs. Otherwise, any editor supported by the operating system can be used. Editors come in different forms. Some editors allow only editing at the level of line i.e., a line can be added and a line can be deleted not individual characters. Nowadays almost all the editors allow insertion and deletion of characters.

The loader is a system program which loads the executable file into primary memory for execution. Sometimes loading involves allocating storage space. Apart from loading the program, any parameters are to be stored onto the stack, and control has to be transferred to the first statement of execution. For convenience primary memory is divided into three parts: text part which can store both programs and data, stack part which stores return addresses and parameters, and heap part which is used for dynamic allocation of memory. The entire program has to be present in the primary memory for execution. If the program size is small it is achievable. If the program size is larger than the primary memory then dynamic loading is used for the execution of programs.

Before execution of the program, it has to be translated into object code. The compilation of the program results in the creation of object (.obj) files. The object files cannot be executed directly. All the required object files are to be linked. Particularly the object file of the source file (program) has to be linked with object files of various libraries before it is executed. Linking can be done at compile time or load time or execution time. If the linking is done at compile time it is called static linking. Some loaders are capable of performing linking also. If the linking is done at execution time it is called dynamic linking. The advantage of having linker is different source programs can be translated to object code separately and then executed by linking them.

Programming languages can be classified into following types: procedure-oriented, object-oriented, functional, logic and scripting languages. In procedure-oriented

programming languages, the main concentration is on the function (procedure) to be performed by the program. An example is C language. Object-oriented programming languages are data-centric. The data and the operations that operate on the data are brought together using the principle of encapsulation. Examples are C++ and JAVA. Functional programming languages achieve the primary result due to the computation of functions whereas procedure-oriented programming languages achieve their primary result due to change of state of variables. An example is LISP. Logic programming encourages the use of mathematical logic for programming. An example is Prolog. Scripting languages are used basically for validation purpose and they are used for gluing things together. An example is Javascript.

Apart from the above classification, the languages can be classified into machine language, assembly language, and high-level language. Machine language varies from computer to computer. The program written in machine language is not portable. The program in machine language is a sequence of 1's and 0's. These programs are difficult to read, understand and debug, but they are efficient. In Assembly language, symbolic codes like ADD, MULT etc are used to write the programs. The programs are readable, but not portable. The assembly language varies from computer to computer. A high-level language is so called because the programs written in a high-level language are independent of the underlying machine and the programs are portable. It is easy to read and understand the programs. The programs take more time to execute.

A program written in a high-level language has to be converted into object code. For this, a translator is required. There are two translators: Compiler and interpreter. The compiler is a translator which converts a program written in the high-level language into object code. The compiler acts on all instructions at a time. A compiler basically checks for any syntax errors. The interpreter is a program which interprets the program one line at a time. The interpreter may convert the program into some intermediate code before interpreting or directly interpret the source language statements. The differences between compiler and interpreter are as follows:

Issue	Compiler	Interpreter
Translation method	Acts on all instructions at a time	Acts on one instruction at a time
Size	Relatively large in size i.e., occupies more space	Relatively small in size i.e., occupies less space
Speed	Relatively Fast	Relatively Slow
Efficiency	Code produced is efficient	Code produced is less efficient
Conversion	Converts to object code	Converts to intermediate code or interprets source language statements directly
Types of languages	Procedure-oriented and object-oriented	Artificial intelligence related
Examples	C, C++	Lisp, Prolog

JAVA is one language which uses both interpreter and compiler. Java uses the compiler to convert the source code to bytecode which is architectural neutral and is portable across different computers. The interpretation of the bytecode requires an interpreter. This is technically called as JAVA Virtual Machine (JVM). Each computer requires its own JVM i.e., JVM's are different for different machines.

A debugger is a program that is used to find program errors (bugs) in the program. A program can be debugged by just using the brute force method (inserting write statements in the program) or sophisticated debuggers can be used. Debugging helps in finding logical errors in the program.

Assemblers are part of the compiler and also assemblers are available separately. Examples are Microsoft assembler (MASM) and Turbo Assembler (TASM). Assembler converts assembly language program into machine code.

1.9.2 Application Software

Application software allows end users to accomplish one or more specific tasks. Typical applications include:

- Game playing
- Communication
- Databases
- Word processors
- Spreadsheets
- Graphics software
- Expert systems

The field of computer market which is fetching a lot of revenues and towards which a maximum number of users especially children got attracted is game playing. A large number of games are freely downloadable. Users spend a lot of time playing games. The games can be played online i.e., without downloading or offline i.e., after downloading.

Communication software allows users to interact with other users and computers to interact with other computers. The trend is using resources wherever available i.e., sharing the resources. This requires communication software which controls the communication process. Users can use communication software to access the Internet. Voice calls can be transferred using communication software and Voice over IP (VOIP) protocol.

Information has become the crucial component of our daily life. When the information is small in size the information can be stored in files of the operating system. But, when the information size increases maintaining the information becomes a complex task. Hence the concept of databases is introduced. The database is a logical collection of information brought together. For processing information, a Database management system is designed that helps the users in a number of ways.

The trend is towards the use of advanced techniques like Data warehousing and Data mining for storing and maintenance of data.

The method of writing letter manually is obsolete. Computers support a variety of word processors which allow the user to type the text easily. Word processor provides many other features like spell checking, changing the font size and color, word art, embedding pictures into the text, formatting the text, etc. Microsoft 'MS-Word' is one of the popular word processing applications. Apart from word processing, the computer has eased the task of accounting. Accounting involves not only typing a lot of text but also the performance of a number of calculations. Spreadsheet applications (an example is 'MS-excel') allow a normal user to perform calculations easily.

Pictures are worth thousands of words. Normal users can prepare their own pictures and also, more importantly, edit the pictures as per their requirement. Editing the pictures is a relatively complex task and requires some practice. Adobe Photoshop is one of the prominent picture processing software.

A computer can act as an expert in any field. This achieved by the design of an expert system. An expert system is a software application designed for a specific purpose. For example, medical diagnosis expert system can prescribe medicines by taking symptoms related input from the users. Expert systems are based on Artificial intelligence.

There are various other application software like automation of manufacturing line, computer-aided design, computer-aided manufacturing, air traffic control, education software, weather forecasting, defense and national security software, etc.

1.10 Types of Computers

Personal Computer/Desktop Computer

A personal computer (PC) is any general-purpose computer whose size, capabilities, and cost make it useful for individuals. Usually, the personal computer is used by a single person for his own personal use. A personal computer is otherwise called as Desktop computer since it can be placed on a desktop. A PC may be used at home or in an office. In office environment to use the PC effectively, it is connected to a local area network (LAN). DELL, IBM, Wipro, etc., are manufacturers of personal computers.

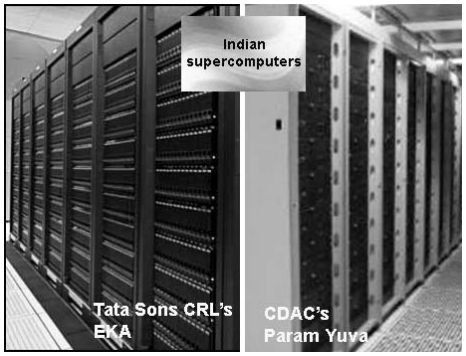
Workstation

A workstation is a high-end personal computer designed for technical or scientific applications. They have high processing capabilities and more memory when compared to PCs. One good example of workstations is SUN Microsystems workstation.

Mainframe Computer

Mainframes are computers which are bulky in size and which were developed before the advent of personal computers. Mainframe computers support multiuser environment i.e., number of users can use the mainframe simultaneously. Each user has the terminal (keyboard and monitor) under his control connected to the mainframe. When the user types the characters, they are sent to the mainframe and then echoed back to the monitor for display. They also have high processing capability and a large amount of main memory which make them suitable for complex and highly critical applications. Some of the applications are Enterprise resource planning, weather forecasting, and stock markets online transactions. Examples of early mainframe system are ENIAC, UNIVAC, and IBM 360. Examples of modern mainframes are zSeries, System z9, and System z10 servers.

Super Computer



A supercomputer is a highly sophisticated computer. It has the highest processing capability and also a huge amount of memory. There is a race in the supercomputer manufacturing and major companies like IBM, CRAY, and HP are in the race. Supercomputers were introduced to the world by Control data corporation (CDC) and led the market into the 1970s until Cray left to form his own company, Cray Research. CRAY continued to lead the race for some time. Most of the supercomputers are custom built i.e., designed for a particular application. Supercomputers use multithreading technology and are highly reliable, scalable and robust computers. The supercomputer is used for highly calculation-intensive tasks such as problems involving quantum mechanical physics, Weather forecasting, climate research, simulation of nuclear weapons, and cryptanalysis. At present Tanhe-1, meaning Milky way is the fastest supercomputer, it achieves a computing speed of 2,570 trillion calculations per second. Please note that by the time the book is published, there may be better super computers.

Nettop

A Nettop is a type of mini desktop computer which performs basic tasks. It is primarily designed to get connected to the computer network and access the Internet, hence the name Nettop. A thin client is the other name for Nettop. It is called a client because it depends on the server for most of the processing and also called a thin client because it has less of everything a computer requires. Nettops cost less, occupies less space and also consume less power; some do not have an optical disk drive and use a solid state disk, making them completely silent. Some of the nettops may not have a hard disk. Nettops are slowly gaining the market share. They can be placed on the desktop or can be attached to back of the LCD monitor.

The computer systems which are cheapest are single unit PCs or Dot stations. Intel is the major manufacturer of them. They combine the monitor and case of the computer within a single unit. Usually, they are in the form of CRT monitor with all the subassemblies within the monitor only. They are not designed for personal use, but basically to connect to the network and then use the resources of the server to which they are connected.

Laptop

A laptop is a small personal computer which is easily portable. They can be carried from one location to other location easily. Though it is a personal computer it is basically used for giving presentations. Relatively its processing capabilities are less when compared to personal computers. Usually, all of the interface hardware needed to operate the laptop, such as parallel and serial ports, graphics card, sound channel, etc., are built into a motherboard. But one of the main problems with it is, it is difficult to upgrade a laptop because most of the functionality is on the motherboard.

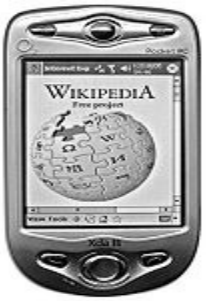
Tablet PC

A Tablet PC is a notebook or slate-shaped mobile computer, first introduced by Pen computing in the early 90s with their PenGo Tablet Computer and popularized by Microsoft. A pen support is provided and a Tablet PC can recognize handwriting. A Tablet PC comes with a wireless antenna enabling it to connect to the computer network and access Internet. Though it can support a variety of applications that can be loaded into a personal computer, the lower performance of Tablet PC enables the user to notice the difference in performance. Hence Tablet PCs are considered to be suitable for simple office applications, Internet browsing, an email application.

Ultra-Mobile PC

The ultra-mobile PC is a small laptop computer. It uses touchscreen display. Ultra-mobile PC is lightweight and can easily be carried from one location to other. It has a display of 4-7 inches.

Pocket PC



A pocket PC is a palm-size computer (personal digital assistant, PDA). It is not comparable to a personal computer. It allows users to access Internet, store and retrieve emails, play multimedia, and games. From a technical standpoint, "Pocket PC" is a Microsoft specification that sets various hardware and software requirements for mobile devices bearing the "Pocket PC" label. For instance, any device which is to be classified as a Pocket PC must:

- Run Microsoft windows mobile operating system
- Support various applications loaded in Read only memory
- Support touch screen

Smart Phone and Tablets

Smartphones are the cellular phones with added intelligence. Telephones were initially designed for making voice calls. But today's telephones are capable of transmitting data, audio, and video. Smartphones have all these capabilities.

Instead of a simple control program, smartphones have full-fledged operating systems which are comparable to computers. Google Android, Apple iOS, and Microsoft Windows phone are the popular operating systems on smartphones. Tablets can be considered as equivalent to smartphones with the major advantage being screen size. The screen size is more, typically 7.5" in tablets and is convenient for watching multimedia content. But they are not as portable as smartphones and also smartphone are better for voice calls and camera resolution features. Smartphones also fit in the pocket that makes it very easy to carry. Smartphones and Tablets are Wi-Fi enabled, and Bluetooth enabled. Both support typical text processing, audio, and video applications. The trend is almost all the applications which can be run on computers are moved to smartphones and tablets.

1.11 Computer Computing Techniques

Batch Processing

It is the first computing technique that allowed execution of computer programs. The term batch processing is used because a collection of jobs (programs) are executed without manual intervention. It was used when computing resources were precious and costly. It permitted sharing. It started with mainframe computers and punched cards containing programs and data. It is non-interactive. Input is not given using the keyboard but through punched cards. All the required input data is put in the files before processing and is given as input along with the jobs to be executed. It is suitable for applications which are intrinsically noninteractive and which require more computing time. Some of the modern applications like Antivirus, E-mail, and Disk clean require batch processing. For example Disk clean removes temporary files and it doesn't require interaction making it suitable for batch processing.

Time-Sharing

Human beings are relatively slow when compared to machines. If a single machine is used by a single person, it is an inefficient use of the machine. Most of the time machine remains idle. Time-sharing is sharing a computing resource among many users simultaneously. The computer serves the different users by sharing time i.e., user programs are executed for a short interval of time (time slice), next some other user programs are executed and so on.

Time-sharing started with mainframes. Since mainframes are large computers having high processing capability and large memory, many terminals are connected to it. Terminals are just keyboard and monitor without any processor or hard disk. Each user is given one terminal and time is shared between the users. Users feel that an entire mainframe is under their control due to the relative difference in speed of users and machine. All the keystrokes are transferred to mainframe and results are transferred back and displayed on the monitor. Though personal computers have become relatively cheap, still terminals are used for many applications.

Client/Server Computing

The client is one which makes a service request. The server is one which processes the requests from clients and sends back the results. The distinction between client and server is based on software (the work that is done by the system). But since server services many clients, usually, server machines have a higher configuration (better processor, more memory) when compared to clients. The client and server may be near to each other or separated by several kilometers but connected by a network. Many of the internet applications like e-mail, WWW are based on Client/Server model.

Clients and servers can be classified based on the function they do. Specific types of clients include web browsers, email clients, and online chat clients. Specific types of servers include Web servers, Applications servers, Database servers, Nameservers, Mail servers, and File servers.

Distributed Computing

Distributed Computing is a layer of abstraction created over a computer network. A computer network consists of an interconnected collection of computers. The presence of different computers is not hidden and the users have to explicitly access the services by logging into the specific computer system. In distributed computing the presence of different computers is hidden. It is up to the distributed computing software to provide services wherever they are available. User's access to the services is implicit.

1.12 Software Development Methodology

Software methodology is an approach to software development. The software development should be reliable, efficient and easily maintainable. Traditionally a systematic approach is being used for software development. The steps in the software development can be treated as software development life cycle (SDLC). SDLC involves the use of the following steps in the order given below:

1. Requirements Gathering and Analysis
2. Design
3. Coding (construction)
4. Testing
5. Deployment and Maintenance

Requirement Gathering and Analysis

This is a first fundamental step of software development. The software is developed for a particular purpose. The software has to satisfy the different requirements of the customer. The requirements are to be extracted, analyzed and should be represented in detail form, understandable to both customer and developer.

It is difficult for the customer to clearly state all his requirements. The customer usually gives his general requirements and assumes some of the requirements are implicit and the software developed will satisfy those requirements. But unfortunately, the software developed will not satisfy, alteast some of the implicit requirements. Hence requirements gathering phase allows the developer to clearly understand all the requirements and state them in explicit terms.

There can be few problems even with the explicitly stated customer requirements. There can be contradictory requirements and requirements which are not feasible to satisfy. The contradictory requirements are to be eliminated and the requirements which can be satisfied by the software being developed and which are beyond the scope of the software being developed should be separated. This phase ends with the 'Requirements specification document' which is agreed by both the customer and developer.

Design

The design is considered as the heart of the software development process. If design goes wrong everything goes wrong. The improper design may result in the successful completion of the software development which doesn't satisfy some of the customer requirements. The design involves the Data design, Architectural design, and design of individual modules.

Data design involves the identification of data structures that can be used to represent the data in detail form. This involves representation of the data identified during analysis phase making the data convenient for the coding phase. Data design may result in representing the data in the form of a database when the data is large in size.

Architectural design results in the identification of different modules and the connectivity between the modules. This is represented using structured charts. Next is the design of individual modules where the local data that is used by the module is represented using appropriate data structures and the procedural detail of the module is represented using pseudo code or algorithmic approach.

Coding

Coding involves selection of appropriate programming language for the representation of data structures identified during the design phase and implementing (writing code) the algorithms. The language should be selected such that there is a clear mapping between the data structures identified and data structures supported by the programming language. This is crucial since the program complexity depends on the data structures. If wrong data structures are chosen then writing code becomes a tedious task and error-prone. Choosing appropriate programming language sometimes depends on the customer target environment. If the customer specifies that a program

should be written using a particular programming language then the developer has no choice.

Testing

Software testing is verifying whether the software developed satisfies the customer requirements or not. This involves both functional and nonfunctional requirements. Functional requirements are the functions to be performed by the software resulting in the satisfaction of customer requirements. Nonfunctional requirements are related to performance and timing constraints.

Software testing is intrinsically destructive in nature. In the analysis and design phases, constructive approach is used whereas software is tested with the intent to find errors. Infact testing technique is considered to be a good one if it finds errors. Usually, the software is not tested by the developers but by the third party persons and ultimately by the customer in the customer environment.

Deployment and maintenance

Deployment starts after the code is tested properly. Deployment involves making the software work in the customer environment satisfying all the customer requirements. If the software is developed with a commercial intent to be sold to a large section of the people, the software should be developed to make it work in different environments or atleast should specify the requirements of the environment.

The step of software development which involves performing all of the above steps repeatedly if required is maintenance. Maintenance is required because errors may be found in the developed software which have to be corrected, customer environment may change, the software has to be made adaptable to a new environment and there are changes in the customer requirements or customer requires enhancements to the software. Usually, software maintenance requires some modifications to the customer requirements resulting in modification of the design, coding the required functionally and testing the software.

Case Study

Problem statement: Compute the roots of a quadratic equation ax^2+bx+c.

Requirements gathering and analysis: Analysis varies from problem to problem. When the problem is complex analysis plays a significant role. But for simple problems like the above analysis involves just determining the inputs and outputs, determining the formulas in case of mathematical problems etc.

Input: values of a,b and c
Output: Roots of the quadratic equation and type of the roots.
Formula: Discriminent, $d=b^2- 4*a*c$

Design

Design part involves structured charts and writing the algorithm.

Algorithm

Read the values of a,b and c
Determine the discriminent, $d=b^2-4*a*c$

If (d>0) then
Print ("roots are real")
Root1=(-b+sqrt(d))/(2*a)
Root2 = (-b-sqrt(d))/(2*a)
Print Root1 and Root 2

If (d==0) then
Print ("roots are real and equal")
Root1=Root2 = -b/(2*a)
Print Root1 and Root2

If (d < 0) then
Print ("roots are imaginary")
Realpart = (-b)/(2*a)
Imaginarypart = (sqrt(-d))/(2*a)
Print Realpart and Imaginarypart

Coding

Write the program using a programming language

Testing

Give as input different values of a,b,c and verify the result. All the cases d>0, d==0, and d<0 are to be tested.

Deployment and Maintenance

In this case, it is of little significance.

*** This Case study may be odd to discuss at this point. The user may come back to the case study after studying the remaining sections of the chapter. It is necessary to understand the software development methodology with an example. This is the reason for discussing the case study here. ***

1.13 Top-Down vs Bottom-Up Approaches

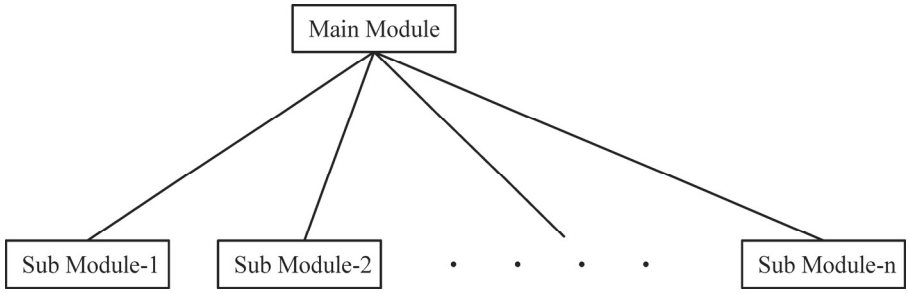
Top-Down and Bottom-Up approaches are two ways of solving the problems. In top-down approach, we start with the main system (main problem) and divide it into a number of subsystems (subproblems) to understand the total system. If necessary

subsystems are further divided until we arrive at the base systems which can be understood easily. In bottom-up approach, we start with subsystems and integrate them to arrive at the main system. The subsystems chosen should be base systems which can be understood easily. In information technology field these two approaches are used to solve the problem. The top-down approach is sometimes called as stepwise refinement. The top-down approach is suitable for all kinds of problems, but it is better if the problem to solve is of moderate or low complexity. As the complexity increases, designers tend to prefer bottom-up approaches. But it is difficult to start with bottom-up approach if the base systems are not clear which is often the case. Usually, Artificial intelligence problems in the fields of neural networks, robotics, and genetic algorithms are solved using bottom-up approaches.

Structured Charts

A structured chart in software engineering shows the breakdown of the entire software project into several manageable modules (sub-functions). It is used to show the hierarchical arrangement of the modules and also shows the interconnectivity among the modules. In the structured chart, the modules which are at the bottom perform the functionality that is expected of the software. Modules at the top are interconnectivity modules or control modules.

Structured charts are represented using rectangles and lines. Each rectangular box represents a module. The names of the modules are written inside the box. Line joins two modules that have an invocation relationship i.e., one module acts as a calling module and other module acts as called module. The functionality of the top module is realized through the implementation of lower level modules connected by lines. The modules which are not connected are independent. While designing a solution to a problem the modules should be made independent as far as possible. Structured charts are drawn as part of the design process of software engineering and are used to represent architectural design.



Structured charts can also be used for understanding the problem itself; it need not be used just to represent the solution to the problem in a hierarchical form. Structured charts may be arrived using top-down or bottom-up approaches.

1.14 Powering up a PC

Before a computer is used for any purpose it has to be switched on. As soon as it is switched on the computer prepares itself to accept the user requests. For this, it does various internal checks. The following is the process of Powering up a PC:

1. Press the "Power" button on the computer and the monitor. As soon as the power button is pressed, an LED indicator glows to indicate that power supply is working properly. If the indicator doesn't glow it indicates a problem with the power supply itself.
2. As soon as the power supply is given to the computer, BIOS (basic input output system) takes control of the computer. BIOS do the power-on-self-test (POST). As the name indicates it is a self-test i.e., the computer itself determines whether all its components are working properly or not. The following are some of the tasks performed by BIOS during start-up of a computer:
   A. The BIOS determines whether the video card is functioning properly or not. Usually, the graphics cards come with their own processor and memory. These are initialized.
   B. The BIOS checks to see if this is a cold boot (normal start-up) or a reboot (reset). It does this by checking the value at memory address 0000:0472. A value of 1234h indicates a reboot, in which case the BIOS skip the rest of POST. Any other value is considered a cold boot.
   C. In case of normal boot-up, BIOS verifies RAM by performing a read/write test of each memory address. It checks for a keyboard and a mouse. Usually, an indicator glows on keyboard and monitor. It looks for an expansion bus and, if it finds one, checks all the connected cards. If the BIOS finds any errors during the POST, it notifies with a series of beeps. The beeps can be used to determine the type of the problem. The beeps differ with different manufacturers of computers. Some manufacturers also use different diagnosis instead of using beeps. The following are some of the beep codes:

1-4 short beeps	:	RAM failure
5 Short	:	Processor failure
6 short	:	Keyboard failure
7 short	:	Bad CPU or Motherboard
8 Short	:	Display problem

The BIOS displays details about the system. This typically includes information about the following:

- Processor
- Floppy and Hard disk drives
- Memory
- BIOS revision and date
- Display

The BIOS looks at the sequence of storage devices identified as boot devices in the complementary metal-oxide-semiconductor (CMOS) setup. The boot is short for "bootstrap". Boot refers to the process of launching the operating system. The BIOS tries to initiate the boot sequence from the first device using the bootstrap. The bootstrap loader loads the operating system into memory and allows it to begin operation. It does this by setting up the divisions of memory that hold the operating system, user information, and applications. The bootstrap loader then establishes the data structures that are used to communicate within and between the sub-systems and applications of the computer. Finally, it turns control of the computer over to the operating system.

1.15 Peripheral Devices

A peripheral device is a kind of I/O device which is external to the computer i.e., it is connected to the computer, but it is not placed inside the computer casing. Some devices are exclusively used for input, some exclusively used for output and some devices can be used for both input and output.

Some common input devices include:

Keyboard	-	For typing text
Mouse	-	A Pointing device for selecting items on the screen
Joystick	-	For Game playing
Pen tablet	-	For doing digital signatures etc
Scanner	-	For Scanning photos and text documents
Digital Camera	-	For transferring photos from camera to computer for display and for further processing.
Microphone	-	For giving input to the computer through voice

Some common output devices include:

Monitor	-	For displaying text, images and video
Projector	-	For presentation to a large audience
TV screen	-	For displaying of what is transmitted
Printer	-	For Printing documents, and images etc
Plotter	-	For drawing figures
Speakers	-	For outputting voice, and audio

Some common input and output devices include:

External hard drives	-	For storing & retrieving all kinds of data
Pen drives	-	For storing & retrieving all kinds of data
Digital mixers	-	For mixing information in digital form.

Mixing is generally used for mixing audio during cultural programmes etc.

1.16 Example Problems

Example 1: Exchange of water in two glasses which are full.

Assume there are two glasses which are of equal size and filled with water. You are asked to exchange the water in two glasses.

How do you do it? You cannot pour water in one glass into other glass because the two glasses are already full.

What do you do? You take another (third) glass. Pour first glass water into a third glass, next pour second glass water into first glass, and then pour third glass water into a second glass. This is the solution.

Example 2: Finding the maximum among four pens.

This problem may be solved in different ways. The simple solution is: first compare the first and second pen and find the maximum among them, let it be maximum1. Next, find the maximum among the third and fourth pens, let it be maximum2. Next, compare maximum1 and maximum2 and find the maximum among them which will be the maximum among the four pens.

Example 3: Addition of two numbers.

Suppose you are asked to find the sum of two numbers.

What do you do? You need to know what are the two numbers that are to be added. Let the two numbers be a and b (When you are solving this problem using computers, you have to imagine yourself as a computer. Since the computer has to add two numbers, the computer should ask what are the two numbers). You ask what are the values of a and b. So, you read the values of two numbers and then add them to find their sum.

If we assume that you need to store the addition of two numbers for further use, then we will assume that there is another symbol called c where we store the addition of two numbers. Essentially you perform the operation c = a + b.

Example 4: Finding the maximum among n pens.

You are given n pens and asked to find the maximum among them. How do you do it? Using the procedure similar to that mentioned in example2 is tedious.

The procedure we may adopt is: First let us assume that the first pen is the maximum. Next, compare the maximum with the second pen. If the second pen is bigger than the maximum, make the maximum as the second pen, otherwise maximum remains the same. Next, compare the maximum with the next pen and the process is repeated and all the n pens are compared. By following this procedure we get the maximum among n pens in an identifier called maximum.

Example 5: Check whether the given number is prime or not.

What is a prime number? A prime number is one which is only divisible by one and also by itself. Any number whether it is prime or not is divisible by one and by itself. Hence we need not again check whether the given number (say n) is divisible by one or by itself. We have to bother about only other numbers which can divide n. If there is any such number that divides n, then n is not a prime number.

What are the numbers with which we have to divide n to check for divisibility? One may start dividing from 2 to n-1. No problem this works. But the number of divisions is large. Assume you are given a large number, say 100000. The number of divisions is 99998. It is inefficient. Careful thought reveals that there cannot be any number which is greater than n/2 which can divide n. Hence we can check with numbers in range 2 to n/2.

Further improvement is if there is a number say p which exactly divides n, then there will be a factor less than or equal to squareroot(p). Hence we can check from 2 to squareroot(n) for divisibility of n.

Example 6: Check whether the given string is palindrome or not.

A palindrome is a string which reads same from left to right and right to left. Start comparing the characters at the beginning and at the end i.e., first and last characters. If they are different then string cannot be a palindrome. Otherwise, continue the process with the second character and last but one character. At each step, if the characters differ, the string cannot be palindrome and there is no need to check further. The process is continued only if the two characters match. The process is repeated until the middle character or two middle characters are encountered. In the first case string is a palindrome. In the second case if the two middle characters are same then the string is a palindrome.

Example 7: Searching for a given number in the list of numbers.

There are two cases, the number that is searched is present in the list of numbers or the number is not present. We have to basically compare the search number with each number in the list. In either case, we have to search until the search number is present or all the numbers are compared and we reached the end of the list. If we have reached the end of the list without finding the number then the search number is not present. We start searching by comparing the search number with the first number. If they match the search number is found else proceed comparing with the next number (if any) in the list.

Example 8: Sort the list of numbers.

Assume the numbers are to be arranged in the ascending order. A simple solution is, do n-1 passes over the list of numbers. At each pass, find the minimum among the list

of numbers remaining. Place the minimum number at the beginning. The existing number at the beginning is placed in the minimum position. In the first pass, all the numbers are considered. In the second pass, n-1 numbers excluding the first number are considered since the minimum number is at the beginning and there is no need to consider it. In the next pass, n-2 numbers excluding the first and second numbers are considered. The process is repeated until there is only one number remaining which is in the last position. This results in a sorted list.

1.17 Representation of a Solution to the Problem

There are three ways of representing a solution to a problem: Pseudo code, Algorithm, and Flowchart.

1.17.1 Pseudocode

Pseudocode, as the name indicates, is not the actual code that can be executed by a computer, but a detailed yet readable description of the procedure to solve a problem written in a natural language like English. There is no specific notation for writing pseudocode. But as far as possible it should be general and independent of any programming language. Pseudocode typically omits details that are not essential for human understanding of the algorithm, such as variable declarations, system-specific code, and subroutines.

Usually, programming starts with a pseudocode. It is difficult to write a complex program directly. Hence the main logic of the program is written using pseudocode and then analyzed to find whether it is correct or not. Given the pseudocode it is easy to write a program. Pseudocode is otherwise called as a conceptual algorithm.

Eg:

Pseudocode that obtains two integer numbers from the user and prints the sum of those numbers.

Pseudocode:

Prompt the user to enter the first integer

Prompt the user to enter a second integer

Compute the sum of the two user inputs

Display the result

1.17.2 Algorithm

The word “Algorithm” comes from the Persian author Abdullah Jafar Muhammad ibn Musa Al-kawarimi in the ninth century, who has given the definition of an algorithm as follows:

- "An Algorithm is a set of rules for carrying out calculation either by hand or on a machine".
- "An Algorithm is a well defined computational procedure that takes input and produces output".
- "An Algorithm is a finite sequence of instructions or steps (i.e. inputs) to achieve some particular output".

A program may be thought of as an algorithm expressed in a programming language. An algorithm is independent of any programming language.

An Algorithm is a sequence of steps to be executed in order to solve a problem. An algorithm is independent of any programming language in which the algorithm will be finally implemented. The notation should be as general as possible. An algorithm can be written basically at two levels: Conceptual level and Implementation level. Conceptual level means algorithm is described using general natural language statements explaining only the core of the algorithm, Implementation level algorithm is written with a notation that can easily be translated into programming language statements. But, both are independent of a programming language.

The following are the essential characteristics of an algorithm, proposed by D. E. Knuth, a famous mathematician:

1. **Finiteness:** An algorithm should have a fixed (finite) number of steps.
2. **Definiteness:** Each step of the algorithm should be defined precisely. There should be no ambiguity.
3. **Effectiveness:** All the operations in the algorithm should be basic and be performed within the time limit.
4. **Input:** An algorithm may accept some inputs, but inputs are not mandatory.
5. **Output:** An algorithm should yield one or more outputs.

Algorithm Notation

The framework of the algorithm is as follows:

Algorithm name of the algorithm: problem statement/Aim

Input: input variables with brief description, if any

Output: output variables with brief description

Data Structures: Any specific data structures apart from input and output variables

Begin

Statements solving the problem

End

Other Constructs

Bracketing symbols for enclosing statements: Begin End

Assignment operator: ←

Arithmetic operators:	+		-	*	/	mod
Logical operators:	and	or	not			
Relational operators:	<	>	<=	>=	≠	=

List

The list of numbers can be stored in a List[1..n], where List is the sequence of numbers. The first number is stored at List[1], the second number is stored at List[2], ... the last number is stored at List[n].

Loops

For index variable ← Initial value **to** Final value **increment/decrement by** value

Begin .....statements..... **End**

Initial condition using index variable

While (terminating condition using index variable **)**

Begin …… statements and Increment/decrement of index variable ….. **End**

Conditional statement

If (condition1**) then**

Statement1

Else if (condition2**)**

Statement2

.

.

.

Else

Statement

Input statement: **Read** v1,v2, … vn

Output statement: **Print** v1,v2,v3…..vn

Print "message to be printed"

Example Algorithms

1. Problem: Write algorithm for adding two numbers.

Algorithm Add Two: Addition of two numbers

Input: Two numbers, say a and b

Output: Sum of the two numbers, say c
Begin
Read a
Read b
C ← a + b
Print c
End

---

2. Problem: Write algorithm for exchange of two numbers.

Algorithm Exchange: Exchanging values of two numbers

Input: Two numbers, say a and b

Output: Values of two numbers a and b before and after exchange

```
Begin
        Print "enter the values of two numbers a and b"
        Read a,b
        Print "Values of a and b before exchange"
        Print a,b
        temp ← a
        a ← b
        b ← temp
        print "values of two numbers after exchange"
        print a,b
End
```

Note*: *** *Try without using a temporary variable* *****

---

3. Problem: Find the maximum of two numbers.

Algorithm maximum: The following algorithm finds the maximum of two numbers

Input: Two numbers, say a and b

Output: maximum of two numbers, a or b

```
Begin
        Print "enter the values of two numbers"
        Read a,b
        If (a > b)then
```

```
            Begin
                Print "the maximum of two numbers is"
                Print a
    End
        Else
Begin
        Print " the maximum of two numbers is"
        Print b
    End
End
```

4. Problem: Find the maximum of three numbers.

Algorithm maximum: The following algorithm finds the maximum of three numbers

```
Input: Three numbers, say a, b, and c
Output: maximum of three numbers, a or b or c
Begin
    Print "enter the values of three numbers"
    Read a,b,c
    If (a > b)then
        Begin
            If (a > c)
                Begin
                    Print "the maximum of the given numbers is"
                    Print a
                End
            Else
                Begin
                    Print "the maximum of the given numbers is"
                    Print c
                End
        End
    Else
        Begin
            If (b > c)
```

```
        Begin
                Print " the maximum of two numbers is"
                Print b
        End
    Else
      Begin
                Print "the maximum of two numbers is"
                Print c
        End
End
```

4.1 Problem: The above algorithm can also be written as follows:

```
Algorithm maximum: The following algorithm finds the maximum of three
numbers
Input: Three numbers, say a, b, and c
Output: maximum of three numbers, a or b or c
Begin
    Print "enter the values of three numbers"
    Read a,b,c
    If (a > b) and (a > c)
    then
       Begin
                Print "the maximum of the given numbers is"
                Print a
      End
    Else if (b > a) and (b > c) then
       Begin
                Print "the maximum of the given numbers is"
                Print b
      End
    Else
      Begin
                Print " the maximum of two numbers is"
                Print c
      End
End
```

Conceptual Vs Implementation level Algorithm

The algorithms can be written at two levels, conceptual level, and implementation level. Conceptual level algorithm presents the gist of the procedure to be followed in solving the problem. Implementation level algorithm, though language independent, can be implemented easily. The following example shows the difference between them. The algorithm is to find the biggest number among the given list of numbers.

Conceptual level algorithm:

1. Assume that the first item is the biggest number. Store it in say big-element
2. Look at each of the remaining items in the list. At each step, if the item in the list is bigger than the big-element then store it in big-element
3. The last noted item in the big-element is the biggest number in the list of numbers.
4. Print big-element.

Implementation level algorithm: The following is more close to implementation level but still independent of any programming language.

Algorithm Biggest Number: The following algorithm determines the biggest number in the given list of numbers.

Input: A list of numbers L.

Output: The biggest number in the list L.

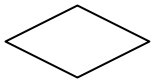
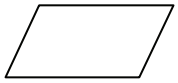
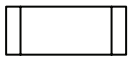
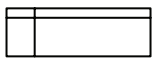
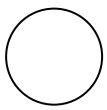
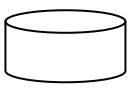
```
Begin
  biggest ← L[0]                    ; the first element in the list
  for each item in the list L
    Begin
        if the item > biggest, then
            biggest ← item
    End
  print biggest
  End
```

1.17.3 Flowchart

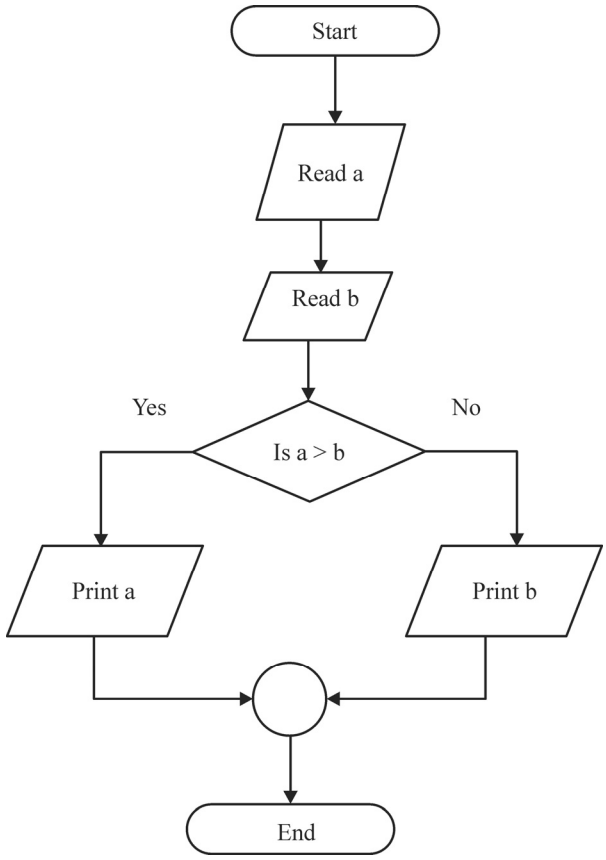
A flowchart is a pictorial representation of the steps in solving a problem. In flowchart steps are represented as boxes of different shapes. The sequence in which different steps are carried out along with the alternatives is represented by connecting boxes with arrows. Flowcharts are used in analyzing, designing, documenting or managing a

process or program in various fields. Similar to algorithm and pseudo code, flowchart is independent of any programming language.

Flow Chart Representation

Flow chart symbol	Name	Description
	Process	An action or operation
	Start/End	Start or End of the Flow Chart
	Decision	Branching based on Decision making
	Input/Output	Input and Output of a process
	Predefined Process	Invocation of already defined process
	Internal Storage	Data stored in memory
	Document	Documentation
	Connector	Joins flows
	Magnetic Disk	Major Type of storage device
	Flow Line	Indicates direction of flow

Eg: Flowchart to find the maximum of two numbers



1.18 Developing a Computer Program

The steps in writing a computer program are:

1. *Define the problem*: Examine the problem until you understand it thoroughly. A clear problem statement is a primary requirement. For simple problems, the problem statement is straightforward. For complex problems, details related to the problem are to be mentioned. Define the scope of the problem. Formulate the problem
2. *Gather the required information for solving the problem:* Any problem requires sufficient information to solve it. The information includes the facts and also assumptions if any.
3. *Develop the evaluation criteria for the solution*: A criterion is a standard, rule, or test by which you can judge something. Criterion is based on facts and assumptions. One has to check for the suitability of the solution, its feasibility and its completeness in solving the problem.

4. *Generate all the possible solutions*: All the possible solutions are to be investigated. The solutions which have worked for similar problems may be reused. But it need not be the best solution and innovative solutions are possible which are better than the previous solutions.
5. *Analyze the possible solutions*: Investigate the merits and demerits of all the possible solutions.
6. *Compare the possible solutions*: Select the best solution among all the possible solutions. The evaluation criterion determined previously should be used.
7. *Write the algorithm*: Represent the solution using an algorithm. The algorithm should be independent of the programming language and should use standard notation for representation. The algorithm is the step-by-step procedure that leads to the solution.
8. *Test the algorithm for correctness*: Provide test data and try to work out the problem as the computer would. This is a critical step but one that programmers often forget. It may not be possible to check the algorithms which are large in size manually. Atleast each statement should be verified for its correctness. The modules identified during algorithm design can be tested separately.
9. *Convert the algorithm into a program*: Translate the instructions in the algorithm into a computer program using any programming language. It is not just about the syntax and semantics of the programming language. Logically suitable data structures, control structure and other features of the programming language should be selected judiciously.
10. *Convert the program into object code*: Use translator compiler/interpreter to convert the source language program into object code.
11. *Run the program*: Instruct the computer to execute the program. The process of running the program differs from language to language. One may use integrated environments like Turbo C IDE or may run the program using commands.
12. *Test and Debug the program*: Make sure that the program runs correctly without any errors or bugs. Testing the program involves two aspects, validation, and verification. Validation is giving all possible inputs and checking whether the output generated is correct or not. It is generally what is done by most of the programmers. Verification is ensuring that the program is correct by using mathematics. Verifying the program can be done for small programs, but for large programs it is a tedious task. The testing intent is to find the errors not to determine their absence. Finding the errors and fixing them is called debugging. Debugging can be done in different ways. Brute force method, whereby different print statements are added to the program is normal method followed by most programmers. Single stepping mode by watching the variable values during the execution of the program is another method. The advanced debugging software can be used for large programs.

13. *Document the program clearly*: Documentation can be done at the coding level and also during all the phases of program development. At coding level, comments can be added at appropriate points. All the phases of software development including customer requirements gathering and analysis, design, coding, and testing are to be documented. Documentation is not the activity which has to be considered after coding, but it has to start as soon as the problem is identified and continues throughout the life of the software including maintenance.
14. *Maintenance*: A correct program solves only the current requirements of the customer. Any software will undergo changes. The changes can be due to errors in the program, change in operating environment, enhancements to the program are required, etc. The maintenance may involve repeating all the above steps and is active as long as the software is used.

The Importance of Data Organization

The way the data is logically organized as perceived by the user is important in problem-solving. The complexity of the solution depends on the assumptions about the data organization. Wrong selection of the data organization can lead to a complex solution. For example, if we assume that (names, telephone numbers) are arranged in unsorted (random) order, then searching for a particular person's telephone number is a complex task. If the data is assumed as arranged in the sorted order then the solution is simple. Similar is the case with searching the dictionary. It is better to assume the data organization first before designing the solution to the problem. The data organization can be mapped to the concept of data structures.

1.19 Practical Knowledge

Typical Laptop specifications

S.No.	Item	Description
1	Make and model	To be specified
2	Processor	Intel core I5 processor 4th generation 2.8G Hz
3	RAM	4 GB DDR3 RAM expandable to at least 16 GB
4	HDD	500 GB 5400 rpm SATA hard drive
5	DVD writer	Yes
6	Graphics	Intel HD Graphics
7	Display	15.6 inch LED display
8	Ethernet	10/100 Mbps RJ-45 Ethernet port

Table *contd…*

S.No.	Item	Description
9	Battery	Lithium (at least 40 Whr) battery
10	Chipset	Intel chipset
11	Webcam	Yes
12	OS	Preloaded Windows 7/8.1(English)
13	Antivirus	Preloaded antivirus of standard make
14	Wi-Fi enabled	Yes (should support IEEE 802.11 b/g/n)
15	Bluetooth enabled	Yes
16	Warranty	One year onsite warranty

Typical Desktop Specifications

S.No	Item	Description
1	Make and model	To be specified
2	Processor	Intel core I5 processor 4^{th} generation 2.8G Hz
3	RAM	4 GB DDR3 RAM expandable to at least 16 GB
4	HDD	500 GB 7200 rpm SATA hard drive
5	DVD writer	Yes
6	Graphics	Intel HD Graphics
7	Display	15.6 " LCD monitor
8	Ethernet	10/100 Mbps RJ-45 Ethernet port
9	Chipset	Intel chipset
10	OS	Preloaded Windows 7/8.1(English)
11	Wi-Fi enabled	Yes (wireless LAN card built-in or as expandable card, should support IEEE 802.11 b/g/n)
12	Bluetooth enabled	Yes
13	Keyboard	Standard keyboard
14	Mouse	Optical mouse
15	Warranty	One year onsite warranty

Typical Server Specifications

S.No	Item	Description
1	Make and model	To be specified
2	Processor	Two Intel Xeon V3 processors
3	RAM	8 GB DDR4 RAM expandable to at least 32 GB
4	HDD	Two 4 TB SAS drives, Four 2 TB SATA drives
5	DVD writer	Yes
6	Graphics	Intel HD Graphics
7	Display	15.6 “ (or higher) LCD monitor
8	Ethernet	10/100/100 Mbps RJ-45 Ethernet ports four in number
9	Chipset	Intel C610 series chipset
10	OS	Preloaded windows server 2012 SP2 (English) 64 bit
11	Wi-Fi enabled	Yes (wireless LAN card built-in or as expandable card, should support IEEE 802.11 b/g)
12	Bluetooth enabled	Yes
13	Keyboard	Standard keyboard
14	Mouse	Optical mouse
15	RAID Support	Yes, RAID 5
16	Warranty	Three years onsite warranty
17	PCI Slots	8 in number
18	Mount	Tower mount

Typical Printer Specifications

S.No	Item	Description
1	Cartridge	12A
2	Colour	B/W
3	Main input tray	100 sheets of regular weight
4	Paper types	A4, Legal, Letter
5	Print Speed	12 pages per minute with the first page out in 10 seconds
6	Print Resolution	600 x 600 dpi

1.20 Number Systems

Computers are basically number processing devices. Everything inside the computer system is represented using numbers. There are different types of number systems. Some of the examples are decimal, binary, octal, and hexadecimal. There are other types of number systems such as Roman number system. But since computers don't use this number system we don't consider it here.

The number systems are positional systems. Few symbols are supported by the number systems. These symbols are combined to form the specific numbers of the number system. The position of the symbol determines its value. A multiplication factor is associated with each position. Usually, the rightmost symbol has least multiplication factor. As we move left the multiplication factor value increases.

The multiplication factor is nothing but the base of the number system, say p. The rightmost positioned symbol is multiplied by p^0 i.e., 1 (unity). The next symbol is multiplied by p^1, the next symbol by p^2 and so on.

If xyz is a valid number in the number system of base p, then its value is

$x \times p^2 + y \times p^1 + z \times p^0$

Number systems are classified based on the base of the number system. If the base is 10 then the number system is called decimal number system, if the base is 2 then the number system is called binary number system, if the base is 8 then the number system is called octal number system, and if the base is 16 then the number system is called hexadecimal.

In the decimal number system the only possible digits are 0,1,2,3,4,5,6,7,8,9. These digits can be combined to form valid numbers. Examples are 10, 234, 5467 etc.

If we consider 234 as the number, then 4 is at units place, 3 is at tens place and 2 is at hundreds place. The value of 234 is computed as follows $2 * 10^2 + 3 * 10^1 + 4 * 10^0$ i.e. Two hundred and thirty four.

In the binary number system, the only possible symbols are 0 and 1. This is the reason for calling it as a binary number system. Example of a binary number is 101101.

Any value in a number system can be converted to an equivalent value in other number systems. Generally, we convert the values to decimal number systems.

If we consider 101101 as the binary number, the value of it in decimal is

$$1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 32 + 0 + 8 + 4 + 0 + 1 = 45$$

In the octal number system, the only possible symbols are 0, 1, 2, 3, 4, 5, 6, 7, a total of 8 symbols are used and that is the reason for calling it as an octal number system. Example of an octal number is 234.

If we consider 234 as the octal number, the value of it in decimal is

$$2 \times 8^2 + 3 \times 8^1 + 4 \times 8^0 = 128 + 24 + 4 = 156$$

In the hexadecimal number system, the only possible symbols are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 and symbols a, b, c, d, e, f total of 16 symbols are used and that is the reason for calling it as a hexadecimal number system. The values of letters a-f are 10, 11, 12, 13, 14, 15 respectively. Example of an octal number is 23af.

If we consider 23af as the hexadecimal number, the value of it in decimal is

$$2 \times 16^3 + 3 \times 16^2 + a \times 16^1 + f \times 16^0 = 8192 + 768 + 160 + 15 = 9135$$

Commonly used notation for representing number in a particular number system is

$Number_{base}$

Eg: 234_{10}, 234_8, 234_{16}, and 10101_2

General procedure to convert a decimal number to binary/octal/hexadecimal system

1. Take the base of the number system as the divisor and the decimal number as a dividend.
2. Divide the dividend by divisor and note the remainders and quotients at each step.
3. At each step after the second step consider quotient as the new dividend and continue the process of the division until the quotient is zero.
4. Start writing the sequence of remainders obtained with the last remainder on the left side.

   If $x_1\ x_2\ x_3\ \ldots\ x_n$ are the remainders obtained in that sequence then the number is $x_n\ x_{n-1} \ldots\ x_2\ x_1$.

   Eg: Convert decimal 124_{10} to binary number

Divisor	Dividend	Remainder
2	124	0
2	62	0
2	31	1
2	15	1
2	7	1
2	3	1
2	1	1
	0	

The equivalent binary number of decimal 124_{10} is 1111100_2

Eg: Convert decimal 124_{10} to octal number

Divisor	Dividend	Remainder
8	124	4
8	15	7
8	1	1
	0	

The equivalent octal number of decimal 124_{10} is 174_8

Eg: Convert decimal 124_{10} to Hexadecimal number

Divisor	Dividend	Remainder
16	124	12 (c)
16	7	7
	0	

The equivalent hexadecimal number of decimal 124_{10} is $7c_{16}$

1.21 Overview of Microsoft Windows and UNIX/LINUX

Microsoft Windows

One of the most popular operating systems and which has the largest user database is Microsoft Windows. It is not a single operating system but a collection of series of operating systems. The operating systems developed by Microsoft can be broadly classified into two types: Desktop operating systems and Server operating systems. Windows 7 and Windows 8 are the latest desktop operating systems and windows server 2012 is the most popular server operating system. There exist operating systems not only for computers but also for smartphones. Examples being Android (not Microsoft operating system) and windows phone.

A special characteristic of Microsoft operating systems is, from the beginning the operating systems of Microsoft supported Graphical user interface (GUI). UNIX and Linux variants are considered as efficient operating systems. Because of lack of GUI, they were not popular. Hence Microsoft operating systems achieved enormous success and they have huge user share. Though GUI is added to UNIX and Linux variants, still Microsoft has a major market share.

The journey started by creating a layer of GUI abstraction and adding it over the popular Microsoft Disk operating system (MS-DOS). Disk operating system (DOS) was one of the first operating systems designed for computers. It was initially used by IBM personal computers which were popular at that time. But as Microsoft entered the operating systems market, DOS of IBM was relabelled and was called PC-DOS. The

Microsoft version of DOS is called MS-DOS. Initially, the Microsoft operating systems were based on DOS. DOS supported command interpreter for executing commands at the command prompt. Later Microsoft integrated DOS with its GUI operating systems.

The success of the personal computers is one of the main reasons for the success of Microsoft Windows. Among the series of operating systems released by Microsoft, the most popular are Windows 3.1., Window 95, Windows 98, Windows NT, Windows XP, Windows 7, Windows 8.1, Windows Server 2003, Windows Server 2008, and Windows server 2012.

UNIX & LINUX

UNIX is the efficient operating system developed by Ken Thomson and Dennis Ritchie at Bell Labs in the 1970s. It is considered the multiuser and multitasking operating system. Now all the operating systems support multiple users and execute multiple tasks simultaneously. A multiuser operating system means multiple users can access the operating system services simultaneously. Even with a single processor, multiple users can be provided the illusion of total exclusive control over the computer by sharing time. The important point to remember is, a machine is very fast when compared to human and hence we can provide the illusion of total exclusive control. Multitasking is executing more than one program simultaneously in the sense that when one program waits for I/O, CPU can be allocated to another program.

One of the reasons for the UNIX to limit itself to academic and industry environments initially and that helped Microsoft to become more popular is, it supported only command mode i.e., GUI was not supported. Now all the UNIX variants are GUI based. The most popular UNIX operating systems are SCO UNIX and BSD (University of California, Berkeley) UNIX. The UNIX server operating systems are considered as efficient and stable and are more commonly used than Microsoft Servers.

Since UNIX is commercial, a variant of operating systems called Unix-like operating systems was developed and called LINUX named after LINUX Torvalds. There are many variants of LINUX, the popular being RedHat, Fedora, and Ubuntu. These LINUX variants are free and open source software which can be downloaded. The stable and tested variants are also available for nominal cost. It is very popular for servers and other sophisticated computers. The source code of LINUX operating system can be modified by anyone with the condition the modified source code should be made available under GNU General Public License.

1.22 DOS Commands Summary

ASSOC Displays or modifies file extension associations.

AT Schedules commands and programs to run on a computer.

ATTRIB	Displays or changes file attributes.
BREAK	Sets or clears extended CTRL+C checking.
CACLS	Displays or modifies access control lists (ACLs) of files.
CALL	Calls one batch program from another.
CD	Displays the name of or changes the current directory.
CHCP	Displays or sets the active code page number.
CHDIR	Displays the name of or changes the current directory.
CHKDSK	Checks a disk and displays a status report**.**
CHKNTFS	Displays or modifies the checking of disk at boot time**.**
CLS	Clears the screen.
CMD	Starts a new instance of the Windows command interpreter**.**
COLOR	Sets the default console foreground and background colors.
COMP	Compares the contents of two files or sets of files.
COMPACT	Displays or alters the compression of files on NTFS partitions.
CONVERT	Converts FAT volumes to NTFS. You cannot convert the current rive.
COPY	Copies one or more files to another location.
DATE	Displays or sets the date.
DEL	Deletes one or more files.
DIR	Displays a list of files and subdirectories in a directory.
DISKCOMP	Compares the contents of two floppy disks.
DISKCOPY	Copies the contents of one floppy disk to another.
DOSKEY	Edits command lines, recalls Windows commands and creates macros.
ECHO	Displays messages or turns command echoing on or off**.**
ENDLOCAL	Ends localization of environment changes in a batch file.
ERASE	Deletes one or more files.
EXIT	Quits the CMD.EXE program (command interpreter).
FC	Compares two files or sets of files and displays the differences between them.
FIND	Searches for a text string in a file or files.
FINDSTR	Searches for strings in files.
FOR	Runs a specified command for each file in a set of files.
FORMAT	Formats a disk for use with Windows**.**

FTYPE	Displays or modifies file types used in file extension associations.
GOTO	Directs the Windows command interpreter to a labeled line in a batch Program
GRAFTABL	Enables Windows to display an extended character set in graphics mode.
HELP	Provides Help information for Windows commands.
IF	Performs conditional processing in batch programs.
LABEL	Creates, changes, or deletes the volume label of a disk.
MD	Creates a directory.
MKDIR	Creates a directory**.**
MODE	Configures a system device**.**
MORE	Displays output one screen at a time.
MOVE	Moves one or more files from one directory to another directory.
PATH	Displays or sets a search path for executable files**.**
PAUSE	Suspends processing of a batch file and displays a message**.**
POPD	Restores the previous value of the current directory saved by PUSHD**.**
PRINT	Prints a text file.
PROMPT	Changes the Windows command prompt.
PUSHD	Saves the current directory then changes it.
RD	Removes a directory.
RECOVER	Recovers readable information from a bad or defective disk.
REM	Records comments (remarks) in batch files or CONFIG.SYS.
REN	Renames a file or files.
RENAME	Renames a file or files.
REPLACE	Replaces files.
RMDIR	Removes a directory.
SET	Displays, sets, or removes Windows environment variables.
SETLOCAL	Begins localization of environment changes in a batch file.
SHIFT	Shifts the position of replaceable parameters in batch files.
SORT	Sorts input.
START	Starts a separate window to run a specified program or command.
SUBST	Associates a path with a drive letter**.**

TIME	Displays or sets the system time.
TITLE	Sets the window title for a CMD.EXE session**.**
TREE	Graphically displays the directory structure of a drive or path.
TYPE	Displays the contents of a text file.
VER	Displays the Windows version.
VERIFY	Tells Windows whether to verify that your files are written correctly to a disk.
VOL	Displays a disk volume label and serial number**.**
XCOPY	Copies files and directory trees.

1.23 LINUX Commands Summary

Examining Files

cat file..	display contents of the file
more	display file by screenful (BSD)
tail	display last part of the file

Managing Files

cp	copy file
cp file.1 file.2	make copy of "file.1" called "file.2"
ls	list contents of the directory
ls[-ltdru] file..	list contents of directory options: -l long format -t by time of modification -d directory status -r list in reverse order -u by time of last access
mv	move or rename files and directories
mv file.1 file.2	change name of "file.1" to "file.2"
mv file...directory	move one or more files to a specified directory
mv directory.1 directory.2	change name of "directory.1" to "directory.2"
rm	remove files or directories
rm [-ir] file...	remove one or more files or directories options: -i interrogative -r remove specified directories and their contents

Text Editing

vi	invoke the visual editor
vi file...	invokes vi text editor to edit specified file(s) option: -r recover lost file

Formatting Files

pr	prepare a file for printing
pr [option] file	paginate file for printing options: -h change header -t suppress header
nroff	format text
nroff [-mm,-ms] file..	format text according to commands and defaults options: -mm specifies the use of mm macros package -ms specifies the use of ms macros package
checkmm	check mm macro input
checkmm file..	check input files for formatting errors
checknr	check ms macro input (BSD)
checknr file..	check input files for formatting errors

Printing Files

lp	print file online printer
lp [options] file..	puts the specified file(s) in line printer queue options: -c send a copy of the file for printing -m Send mail when file printed -n# make #- number of copies of printout -d destination printer
lpstat	display information about the status of the LP print
cancel	cancel lp command

Communication

mail	send or receive mail among users
mail username	sends electronic mail to specified users
who	lists users currently online
who [am i] [am I	displays login name, tty number, and login time for users currently logged in.
write	write to another user
write username[tty.no]	send lines from your terminal to another. to end message, type ^D.

mesg	permit or deny write messages
mesg [n] [y]	with n argument, prohibit messages sent to you via write with y argument, permission to write to you is restored

Working with Directories

cd	change directory
cd dir.name	establish specified directory as the working directory
cd ..	establish parent of working directory as the working directory
mkdir	create directory
mkdir dir.name..	make one or more directories
pwd	print working directory
pwd	display absolute pathname of the working directory
rmdir	remove directory
rmdir dir.name..	remove specified directories only if empty. Also, see rm -r.

Managing Background Processes

&	detach commands. At end of command line, causes the shell to execute a command in the background, e.g., spell file &.
ps	process status; displays information about each currently active process
kill -9	terminate process
kill-9 process.id..	terminates one or more processes specified by process ID

Account Management

passwd	change your password

On-Line Documentation

help	provides help
help [args]	explains the use of command or meaning of diagnostic
man	display online UNIX manual
man [option][chapter] [title...	prints section for any specified UNIX command

Other Utilities

cal	display calendar
cal [month]	displays calendar for any year or month between 1 and 9999

year	A.D.
cmp	compare two files
cmp file.1 file.2	compares two files: if different, displays line and character number where first difference appears, then stops

chmod — change mode

chmod[who] operator permission file... — changes read, write, and execute privileges for the file for owner, group, others, or all

Who:	u user (owner) g group o others a all of above
Operator:	+ add permission - remove permission
Permissions:	r read w write x execute

date — display date and time

diff — display differences between files

diff[-bh] file.1 file.2 — Compares two files and indicates what lines must be changed to make the first file identical to second:

Options: -b ignore trailing blanks

-h half-hearted option; incapable of handling complex changes

echo — repeat command line arguments

echo argument — displays its arguments

find — search for files in the hierarchy processes each file beneath specified directory pathnames, applying specified directive until it encounters directive whose result is false. Directives:

find pathname..directive..

-atime *n*	true for files last accessed *n* days ago
-exec	execute the specified command
-mtime *n*	true for files last modified *n* days ago
-name file	specify one or more file names

ok*command*	same as -exec except that find asks for yes or no confirmation before acting on the file
-print	display pathname of the file

grep
grep [option].. pattern [file]...

Search file for pattern
Searches for patterns in specified list of files or in standard input

Options:
- -c count option
- -l display only list of file names containing pattern
- -v display all lines not matching the pattern
- -y ignore case

sh
sh file [args]...

Execute shell file
Executes commands read from a shell file

sort
sort [-bdfntu] [+field] [-o] [file] ...

sort or merge files
Sort files Options:

-b	ignore leading blanks
-d	sort in dictionary order
-f	fold upper and lower case characters
-n	order numbers in numerical order
-tx	field separator x
-u	eliminate duplicate lines
-o *file*	send output to the specified file

Field Specifier: Notation +1 instructs sort to use sort key excluding the first field.

spell
spell [options] file ...

Find spelling and typing errors
Check words in file against spell's dictionary
Options: +sup.file refer to the specified supplemental spelling list

tee
tee [-a]

Pipe fitting
Sends input simultaneously to standard output and specified file

file	Options:-a appends to specified file
wc	Word count
wc [file...] ...	Count words, characters, and lines in specified files or standard input

1.24 Sample C Projects

1. Game applications like tic-tac-toe
2. Editors like design of screen editor
3. Compiler implementation (Eg: GCC)
4. Interpreter implementation
5. Design of operating system
6. Calendar application
7. Employee payroll system
8. Student attendance calculation
9. Student class calculation
10. Calculator application
11. Calendar application
12. Digital Clock application
13. Operating system (UNIX was designed using C language)
14. Searching and sorting
15. Library routines related to different languages
16. Antivirus
17. Device drivers
18. Network communication related
19. Hardware interfacing
20. Disk checking
21. Main memory management
22. Secondary memory management
23. Embedded programming (Toys etc)
24. Media players (Eg: VLC media player)
25. Paint application (Eg: TUX Paint)
26. Simulations (Eg: Stellarium)

1.25 Example Algorithms

1. Problem: Find the sum of even and odd numbers separately in the given list of numbers read dynamically.

Algorithm Sumevenodd: The following algorithm finds the sum of even and odd numbers separately in the list of numbers

Input: List of numbers, n

Output: sum of even and odd numbers, sum even, sum odd

Begin

Print “enter how many numbers are there on the list”

Read n

Print “enter the n numbers one by one”

For i ←1 to n increment by 1

Begin

Read x

If (x mode 2 = 0) then

Sumeven ← sumeven + x

Else

Sumodd ← sumodd + x

End

Print sumeven, sumodd

End

---

2. Problem: Find the sum of even and odd numbers separately in the numbers from 1 to n.

Algorithm Sumevenodd: The following algorithm finds the sum of even and odd numbers in the numbers from 1 to n

Input: List of numbers from 1 to n

Output: sum of even and odd numbers, sumeven, sumadd

Begin

Print “enter how many numbers are there in the list”

Read n

For i ←1 to n increment by 1

Begin

If (i mode 2 = 0) then

Sumeven ← sumeven + i

Else

Sumodd ← sumodd + i

```
End
        Print Sumeven, Sumodd
End
```

---

3. Problem: Given the set of n student marks (in the range 0-100), make a count of the number of students who have passed the examination. Pass is awarded for all marks of 50 and above.

Algorithm passfail: The following algorithm finds number of students who have passed/failed in an examination

Input: number of students, say n and marks of n students with each mark represented by m

Output: Number of students who have passed and failed in an examination, pass, fail

```
Begin
        Print "enter the number of students"
        Read n
        For i ← 1 to n increment by 1
Begin
                Read m
                If (m >= 50)
                        pass ← pass + 1
                Else
                        fail ← fail + 1
        End
        Print pass, fail
End
```

---

4. Problem: Given the base salary of an employee, find the gross salary of the employee. Hra is 10% of basic, da is 50% of basic, and IT is 20% of basic.

Algorithm salary: The following algorithm finds gross salary of an employee

Input: basic salary, basic

Output: gross salary, gross

```
Begin
        Print "enter the basic salary of the employee"
        Read basic
        hra ← basic * 0.1
        da ← basic * 0.5
        It ← basic * 0.2
        gross ← basic + hra + da – it
        Print gross
End
```

5. Problem: Find the average marks of students in a subject.

Algorithm average: The following algorithm finds average of marks of students in a particular subject

Input: number of students, say n and marks of individual students

Output: average marks, average

```
Begin
    Print "enter the number of students"
    Read n
    For i ← 1 to n
    Begin
            Read m
            Sum ← sum + m
    End
    Average ← sum / n
    Print average
End
```

6. Problem: Find whether the given number is palindrome or not.

Algorithm numberpalindrome: The following algorithm checks whether the given number is palindrome or not

Input: number, say n

Output: message palindrome or not palindrome

```
Begin
    Print "enter the number"
    Read n
    sum ← 0
    Temp ← n
    While (n ≠ 0)
    Begin
        r ← n mod 10
        sum ← sum * 10 + r
        n ← n / 10
    End
        If (temp = sum)
        Print "palindrome"
    Else
        Print "not-palindrome"
    End
```

7. Problem: Find the sum of the alternate digits of a number.

Algorithm sumalternate: The following algorithm finds the sum of the alternate digits of a number

Input: number, say n

Output: sum of the alternate digits sum1, sum2

```
Begin
    Print "enter the number"
    Read n
    Sum1 ← sum2 ←0
    flag ← 1
    While (n ≠ 0)
    Begin
        r ← n mod 10
        n ← n / 10
        If (flag = 1)
```

```
Begin
            Sum1 ← sum1 + r
            Flag ← 0
        End
        else
        Begin
            flag ← 1
            sum2 ← sum2 + r
        End
End
Print sum1, sum2
End
```

8. Problem: Find the sum of the n terms of an arithmetic progression.

Algorithm sumap: The following algorithm finds the sum of the n terms of an arithmetic progression

Input: a, d, and n

Output: sum of the arithmetic progression

```
Begin
    Print "enter the first term, number of terms and difference"
    Read a,n,d
    Sum ← (n * (2 * a + (n-1) * d)) / 2
    Print sum
End
```

9. Problem: Find the factorial of a given number.

Algorithm factorial: The following algorithm finds the factorial of a given number

Input: n

Output: factorial

```
Begin
    Print "enter the value of n"
    Read n
    Factorial ← 1
    For i ← 1 to n increment by 1
        Factorial ← factorial * i
    Print factorial
End
```

The ENIAC Story

The world's first electronic digital computer was developed by Army Ordnance to compute World War II ballistic firing tables.

As in many other first along the road of technological progress, the stimulus which initiated and sustained the effort that produced the ENIAC (Electronic Numerical Integrator and Computer) the world's first electronic digital computer was provided by the extraordinary demand of war to find the solution to a task of surpassing importance.

The ENIAC was placed in operation at the Moore School, component by component, beginning with the cycling unit and an accumulator in June 1944. This was followed in rapid succession by the initiating unit and function tables in September 1945 and the divider and square-root unit in October 1945. Final assembly took place during the fall of 1945.

By today's standards for electronic computers, the ENIAC was a grotesque monster. Its thirty separate units, plus power supply and forced-air cooling, weighed over thirty tons. Its 19,000 vacuum tubes, 1,500 relays, and hundreds of thousands of resistors, capacitors, and inductors consumed almost 200 kilowatts of electrical power.

But ENIAC was the prototype from which most other modern computers evolved. It embodied almost all the components and concepts of today's high- speed, electronic digital computers. Its designers conceived what has now become standard circuitry such as the gate (logical "and" element), buffer (logical "or" element) and used a modified Eccles-Jordan flip-flop as a logical, high-speed storage-and-control device. The machine's counters and accumulators, with more sophisticated innovations, were made up of combinations of these basic elements.

ENIAC could discriminate the sign of a number, compare quantities for equality, add, subtract, multiply, divide, and extract square roots. ENIAC stored a maximum of twenty 10-digit decimal numbers. Its accumulators combined the functions of an adding machine and storage unit. No central memory unit existed, per se. Storage was localized within the functioning units of the computer.

The primary aim of the designers was to achieve speed by making ENIAC as all-electronic as possible. The only mechanical elements in the final product were actually external to the calculator itself. These were an IBM card reader for input, a card punch for output, and the 1,500 associated relays.

The ENIAC was not originally designed as an internally programmed computer. The program was set up manually by varying switches and cable connections. However, means for altering the program and repeating its iterative steps were built into the master programmer. Digit trays, long racks of coaxial cables, carried the data from one functioning unit to another. Program trays, similarly, transferred instructions; i.e., programs. In purely repetitive calculations the basic computing

sequence was set by hand. The master programmer automatically controlled repetition and changed the sequence as required.

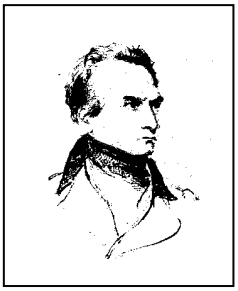
The ENIAC was formally dedicated at the Moore School of Electrical Engineering of the University of Pennsylvania on February 15, 1946, and it was accepted by the U.S. Army Ordnance Corps in July, four years after the original suggestion by Dr. Mauchly.

All During 1946 the ENIAC remained at the Moore School, working out numerical solutions to problems in such fields as atomic energy and ballistic trajectories. Dismantling at the Moore School began in the winter, and the first units arrived at Aberdeen Proving Ground in January 1947. The ENIAC became operational again in August 1947.

The ENIAC's first few years at the Aberdeen Proving Ground were difficult ones for the operating and maintenance crews. The computer represented the largest collection of interconnected electronic circuitry then in existence, and its thousands of components had to remain operational simultaneously. The result was a huge preventive-maintenance and testing program, which, in the end, led to some major modifications of the system.

Those revolutionary modifications, installed early in 1948, converted ENIAC into a serial instruction execution machine with the internal parallel transfer of decimal information.

Charles Babbage Biography



Born December 26, 1791, in Teignmouth, Devonshire UK, Died 1871, London; Known to some as the "Father of Computing" for his contributions to the basic design of the computer through his Analytical machine. His previous Difference Engine was a special purpose device intended for the production of tables.

While he did produce prototypes of portions of the Difference Engine, it was left to Georg and Edvard Schuetz to construct the first working devices to the same design which were successful in limited applications.

Significant Events in His Life

1791 : Born
1810 : Entered Trinity College, Cambridge
1814 : graduated Peterhouse
1817 : received MA from Cambridge
1820 : founded the Analytical Society with Herschel and Peacock
1823 : started work on the Difference Engine through funding from the British Government
1827 : published a table of logarithms from 1 to 108000

1828 : appointed to the Lucasian Chair of Mathematics at Cambridge (never presented a lecture)
1831 : founded the British Association for the Advancement of Science
1832 : published "Economy of Manufactures and Machinery"
1833 : began work on the Analytical Engine
1834 : founded the Statistical Society of London
1864 : published Passages from the Life of a Philosopher
1871 : Died.

In 1822, fed up with the: "intolerable labour and fatiguing monotony" involved in the endless calculation required for scientific tables (a task he believed ranked: "among the lowest occupations of the human intellect" - no argument here!), Babbage came up with a preliminary model of his Difference Engine - a machine, he was sure, that would remove this drudgery forever.

The Royal Society, gathering together to pore over Babbage's crank-driven toothed-wheels and shafts, found the idea: "highly deserving of public encouragement", and convinced the British government to award him £1500 in funding. Babbage confidently predicted his mechanical marvel would be complete within 3 years.

Unfortunately, a working model of the Difference Engine proved considerably more difficult to build than a desktop prototype. Although Babbage spent the next 10 years modifying, enhancing, and re-designing the device - and the British government another £17,000 - in the end, public funding was withdrawn and Babbage was forced to walk away from his brainchild.

A sucker for punishment, by 1833 Babbage had embarked on an even more ambitious project - his Analytical Engine: "a machine of the most general nature". Babbage's Analytical Engine was to be the world's first general use programmable computer, a machine designed not just for solving one particular problem, but to carry out a range of calculations ordered by its operator.

Designed to include both a ‘store’ and a ‘mill’, the Analytical Engine's ‘store’ would retain upto 100 forty-digit numbers awaiting their turn at the ‘mill’. Once operated on, results would also be returned to storage, to be held until needed for further use or printed out. Babbage had quite accurately (if somewhat vaguely) described the modern computer memory and processor.

The Analytical Engine was to be programmed by a series of punched cards, (in much the same way as Konrad Zuse would instruct his Z-series machines in the next century) but would have no ‘inbuilt programming’, or specific purpose, as had the majority of machines designed in Babbage's time.