

Computer Studies



Computer Studies

COMPUTER STUDIES



Website: www.skyfox.co

Email: skyfox@skyfox.org.in

Dr. J. Arockia Venice
Dr. A. Sivaramakrishnan
Mr. Jegan Anthony

Computer Studies

AUTHOR INFORMATION:

Dr. J. Arockia Venice., M.Sc., Ph.D.,

Professor

Department of Computer Science & Engineering

DMI – St. Eugene University

P.O.Box: 330081, Chibombo

Zambia.

Dr. A. Sivaramakrishnan., M.C.A., M.Phil., Ph.D.,

Associate Professor

Department of Computer Science & Engineering

DMI – St. Eugene University

P.O.Box: 330081, Chibombo

Zambia.

Mr. Jegan Anthony., M.C.A., M.Phil.,

Lecturer

Department of Computer Science & Engineering

DMI – St. Eugene University

P.O.Box: 330081, Chibombo

Zambia.

First Electronic Edition : July 2020

Cover Art and Design : Authors

Copyright : © 2020 by Authors

ISBN : 978-81-946968-8-9

DOI : <https://doi.org/10.22573/spg.020.BK/S/003>

[Creative Commons Attribution-Share Alike 4.0 International License](#)

You are free: to Share — to copy, distribute and transmit the work; to make commercial use of the work under the following conditions:

Attribution — you must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that, they endorse you or your use of the work).

No Derivative Works — you may not alter, transform, or build upon this work.

Publisher Contact & Support

Skyfox Publishing Group

#987, Medical College Road

Thanjavur-613004

Tamil Nadu, India.

Phone: +918300123232

Email: skyfoxpublishing@gmail.com / skyfox@skyfox.org.in

Website: www.skyfox.co

Headquarters & Marketing Office

Skyfox Publishing Group

333 Cedar Street, PO Box 208002,

New Haven, United States.

CT 06520-8002.

Tel: 203.737.5603 / Fax: 203.785.7446

Email: skyfoxpublishing@gmail.com / skyfox@skyfox.org.in

Website: www.skyfox.co

Computer Studies

Dr. J. Arockia Venice
Dr. A. Sivaramakrishnan
Mr. Jegan Anthony

From the desk of

Dr. T. X. A. ANANTH, BBA, MSW, MBA, MPhil, PhD,

President – University Council

Dear Learner,

Welcome to DMI – St. Eugene University!

I am sure you are expert in using the PC Tablets distributed by us. Now your world is open to Internet and using the tablet for your educational learning purposes. The very same book you are holding in your hand now is available in your V-Campus portal. All the teaching and learning materials are available in your portal.

As our Chancellor, Rev.Fr. Dr. J. E. Arulraj , mentioned, it is not just the success for DMI-St. Eugene University alone, it is success for the technology, it is success for the great nation of Zambia and it is success for the continent of Africa.

You can feel the improvement in the quality of the learning materials in the printed format. Improvement is done in quality of the content and the printing. Use it extensively and preserve it for your future references. This will help you to understand the subjects in a better way. The theories of the subjects have been explained thoroughly and the problems have been identified at learners' level. This book is made up of five units and every unit is mapped to the syllabus and discussed in detail.

I am happy at the efforts taken by the University in publishing this book not only in printed format, but also in PDF format in the Internet.

With warm regards



Dr. T. X. A. ANANTH

President – University Council

Table of Contents:

S. No.	Topics	Page. No.
1.	Chapter – I: History of Computers Introduction of Computers Memory Peripheral Devices Computer Types	01
2.	Chapter – II: Computer Software Languages Operating systems Microsoft Windows Operating Windows	45
3.	Chapter – III: Data Representation Program Development	73
4.	Chapter – IV: C language Elements of C language Operators, Expressions and Assignment Statements I/O Statements Functions Control statements	92
5.	Chapter – V: Array in “C” language	118
6.	References	138

Chapter I

Computer Hardware: Civic Education of computers Year of invention, name of the inventor and speciality – abacus, Slide rule, arithmetic engine, analytical engine, difference engine, tabulating machine, electronic computer ENIAC, contributions of Pascal, Charles Babbage, Lady Ada Lovelace and Hojn Von Neumann, computer generations – components used in I, II, III, IV and V.

Introduction : Definition of computer, characteristics of computer speed, accuracy, memory, diligence, automation and versatility, hardware, software, block diagram of a personal computer – CPU (Arithmetic, memory and control Units), Input unit, Output Unit, Secondary memory, general idea about PCs.

Memory : Meanings of bit, byte, word, kilo byte, mega byte, giga byte, semiconductor memory – definition and purpose of RAM, ROM, EPROM, EEPROM – typical RAM capacities in personal computers, floppy disks - usage sizes (5.25”, 3.5”) and capacities DSDD, DSHD, 360 KB, 1.2MB, 1.44 MB); organization of floppy disks – usage, capacities, organization of hard disks – sides, cylinders, heads, sectors, CD – ROM drive – usage, capacities and organization.

Peripheral devices (Definition and usage only); input devices – keyboard, mouse, OCR, MICR, output devices – VDU, dot matrix printers, laser printers, ink-jet printers, x – y plotters.

Computer types – Definition, application and examples of analog, digital and hybrid computers; classification of computers based on computing power – micro, mini, mainframe and super computers; network of computers – LAN, WAN, E-mail and Internet.

Chapter II

Computer Software: Languages : **High level language, low level language, assembly language; translators – source program, object program, compiler, interpreter, assembler, comparison of high, low level language and assembly languages; applications of BASIC, FORTRAN, COBOL, PASCAL, C, PROLOG, LOGO and PL/1.**

Utilities and packages : Definition of utilities and packages; list of software’s available for data processing (dBase, FoxPro, RDBMS), word processing (WordStar, WORD, WordPerfect, regional language processor), Spreadsheet, DTP, CAD and bundled software’s (MS – OFFICE ’97).

Operating system: What is an operating system? – mode of operations – Batch, On-line, Time sharing, Real Time, PC operating systems – DOS, UNIX and WINDOWS NT.

Microsoft Windows : What is Windows?, GUI, Functions of Windows, Basic organisation of Windows screens - Windows (types application windows, document windows, group windows), icons, desktop, windows, elements (definition only) – control menu box, title bar, menu bar, maximize button, minimize button and restore button, work space, window borders and pointer; windows organization, list of programs available in MAIN, ACCESSORIES, GAMES and APPLICATIONS group windows.

Operating Windows : Mouse actions – click, double click and drag; start windows; exit windows; open a group window, close a group window; start a program; exit a program. Exercise: Open, edit, save and print document files through windows; use of file manager to change disk drive, view the file details sort files and format a disk.

Internet and Web browsing

Browsers: Mozilla Firefox, Internet Explorer, Netscape Navigator; Control Panel: Add new hardware; Add new Software, Printer Installation, Date/Time, Mouse, and Regional Settings;

Chapter III

Data representation inside the computer:

Binary, Octal and hexadecimal number systems with conversions; Binary Addition and Subtraction – Simple problems; ASCII, BCD and EBCDIC codes.

Program Development:

Problem definition, Algorithm Flowchart, Coding, Compiling & Debugging and Running.

Chapter IV C Language

Elements of C language: Character set, keywords, identifiers, constants, variable, data types, header files and comment statements; declaring, initializing and assigning values to variables.

Operators, expressions and assignment statements: Arithmetic, relational, logical, increment, decrement, conditional, shorthand, assignment operators with hierarchy rules; evaluation of expressions and assignments statements.

I/O Statements: Printf and scanf functions (unformatted and formatted), getchar () and putchar () functions.

Functions: Predefined functions (definition and syntax only) – is digit, isalpha, isupper, is lower and ispunct functions in the header file <ctype.h>; sin, cos, tan, exp, ceil, floor, abs, pow and sqrt functions in the header file <math.h>; strlen, strcpy, strcmp and strcat in the header file <string.h>.

Control Statements

Unconditional goto, if ... else, else if ladder, nested if, switch, for loops, do, while loops and nested loops.

Chapter V Arrays

Defining an Array, Processing Array and Passing Arrays to Functions, Declaring, Initializing and using single, two dimensional arrays.

Chapter I

Aim:

Students will be required to demonstrate knowledge and skills to identify computer hardware, memory, peripheral devices and the different computer types.

Objectives:

- Mastery of the core knowledge of the evolution and introduction to computers.
- Identify the categories of RAM terminology, their locations, and physical characteristics.
- Possession of basic skills on the various types of input/output and classification of computers.

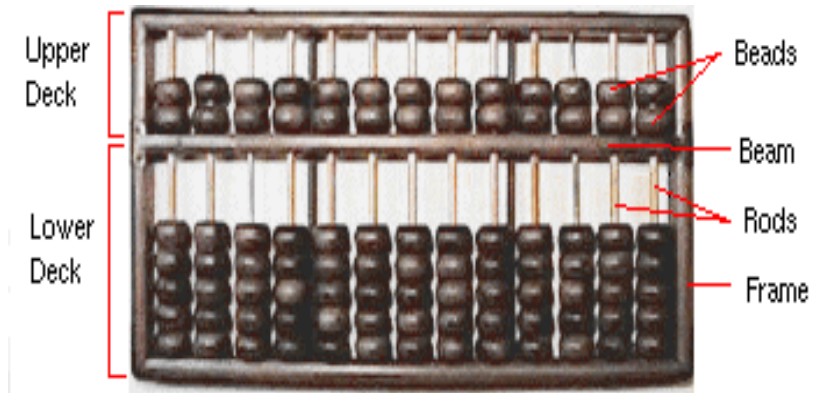
HISTORY OF COMPUTERS:

Computers were invented because of the man's search for fast and accurate calculating devices. The first mechanical adding machine was invented by Blaise Pascal in 1642. Later, in the year 1671, Baron Gottfried Wilhelm von Leibniz of Germany invented the first calculator for multiplication. Keyboard machines originated in the United States around 1880 and are extensively used even today. Around this period only, Herman Hollerith came up with the concept of punched cards, which were extensively used as input medium in computers even in late 1970s. Business machines and calculators made their appearance in Europe and America towards the end of the nineteenth century.

4th Century B.C. ABACUS:

❖ The abacus, a simple counting aid, may have been invented in Babylonia (now Iraq) in the fourth century B.C.

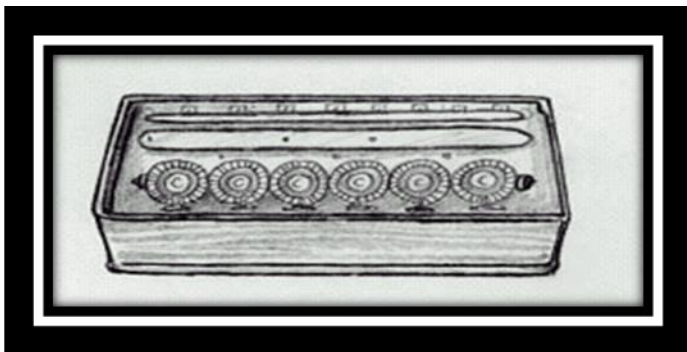
❖ This device allows users to make computation (calculation) using a system of sliding beads arranged on a rack.



BLAISE PASCAL:

Blaise Pascal, he was a French mathematician and experimental Physicist, in 1642 he invented a calculating machine that later came to be called the "Adding Machine. Originally called a "numerical wheel calculator" or the "Pascaline", Pascal's invention utilized a train of 8 moveable dials or cogs to add sums of up to 8 figures long. As one dial turned 10 notches - or a complete revolution - it mechanically turned the next dial.

- Pascal's mechanical Adding Machine automated the process of calculation. Although slow by modern standards, this machine *did* provide a fair degree of accuracy and speed.



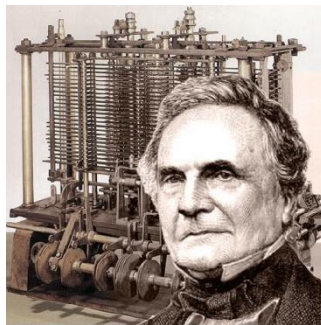
CHARLES BABBAGE (1791 – 1871):

Charles Babbage, a nineteenth century Professor at Cambridge University, is considered the father of modern digital computers. During his period, mathematical and statistical tables were prepared by a group of clerks. Even the utmost (greatest) care and precautions (safety measures) could not eliminate human error. Babbage had to spend a several hours checking these tables. Soon he became dissatisfied and exasperated with this type of monotonous job. The result was that he started thinking to build a machine, which could compute tables guaranteed to be error-free .In this process; Babbage designed a “**Difference Engine**” in the year 1822, which could produce reliable tables. In 1842, Babbage came out with his new idea of **Analytical Engine**, which was intended to be automatic. It was to be capable of performing the basic arithmetic functions for any mathematical problem, and it was to do so at an average speed of 60 additions per minute. Unfortunately, he was unable to produce a working model of this machine, because the precision engineering required to manufacture the machine was not available during that period. However, his efforts established a number of principles, which have been shown to be fundamental to the design of any digital computer.

❖ Born in 1791, Charles Babbage was an English mathematician and professor.

❖ In 1822, he persuaded the British government to finance his design to build a machine that would calculate tables for logarithms.

❖ With *Charles Babbage's* creation of the "**Analytical Engine**", (1833) computers took the form of a general purpose machine.



Charles Babbage

THE EVOLUTION OF COMPUTERS:

The first mechanical adding machine was invented by **Blaise Pascal** in 1642. Later, in the year 1671, Baron Gottfried Wilhelm von Leibniz of Germany invented the first calculator for multiplication. Keyboard machines originated in the United States around 1880 and are extensively used even today. Around this period only, Herman Hollerith came up with the concept of punched cards, which were extensively used as input medium in computers even in late 1970s. Business machines and calculators made their appearance in Europe and America towards the end of the nineteenth century.

1. The **Mark I** Computer (1937-44): Also known as Automatic Sequence Controlled calculator, this was the first fully automatic calculating machine designed by Howard A. Aiken of Harvard University, in collaboration with IBM (International Business Machines) Corporation. Its design was based on the techniques already developed for punched card machinery. It was an electro-mechanical device, since both mechanical and electronic components were used in its design.

2. The Atanasoff-Berry Computer (1939-42). This electronic machine was developed by Dr. John Atanasoff to solve certain mathematical equations. It was called the Atanasoff-Berry Computer, or **ABC**, after its inventor's name and his assistant, Clifford Berry. It used 45 vacuum tubes for internal logic and capacitors for storage.

3. The **ENIAC** (1943-46). The Electronic Numerical Integrator and Calculator (ENIAC) was the first of all electronic computers. It was constructed at the Moore School of Engineering of the University of Pennsylvania in U.S.A by a design team led by Professors J. Prosper Eckert and John Mauchly.

ENIAC was developed because of military need, and was used for many years to solve ballistic problems. It took up the wall space in a 20 X 40 square feet room and used 18,000 vacuum tubes. The addition of two numbers was achieved in 200 microseconds, and multiplication in 2000 microseconds.

4. The **EDVAC** (1946-52):(**Electronic discrete Variable Automatic Computer**) A major drawback of ENIAC was that its programs were wired on boards, which made it difficult to change the programs. This problem was later overcome by the “stored program” concept introduced by Dr. John Von Neumann. The basic idea behind this concept is that a sequence of instructions, as well as data, can be stored in the memory of the computer, for automatically directing the flow of operations. This feature considerably influenced the development of modern digital computers because of the ease with which different programs can be loaded and executed on the same computer. Due to this feature, we often refer to modern digital computers as stored program digital computers. The EDVAC was designed on stored program concept. On Neumann also has a share of the credit for introducing the idea of storing both instructions and data in the binary form (a system that uses only two digits-0 & 1 to represent all characters), instead of the decimal numbers or human readable words.

5. The **EDSAC** (1947-49): **Electronic Delay Storage Automatic Calculator** Almost simultaneously with EDVAC (Electronic discrete Variable Automatic Computer) of U.S.A, the British developed the **Electronic Delay Storage Automatic Calculator** (EDSAC). The machine executed its first program in May 1949. In this machine, addition operation was accomplished in 1500 microseconds, and multiplication operation in 4000 microseconds. The machine was developed by a group of scientists, headed by Professor Maurice Wilkes, at the Cambridge University Mathematical Laboratory.

6. The **UNIVAC I** (1951): The Universal Automatic Computer (UNIVAC) was the first digital computer. Which was not “one of a kind”, Many UNIVAC machines were produced, the first of which was installed in the Census Bureau in 1951 and was used continuously for 10 years. The first business use of a computer, a UNIVAC I, was by General Electric Corporation in 1954.

In 1952, the International Business Machines (IBM) Corporation introduced the 701 commercial computers. In rapid succession, improved models of the UNIVAC I and other 700-series machines

were introduced. In 1953, IBM produced the IBM-650, and sold over 1000 of these computers.

The commercially available digital computers, which could be used for business and scientific applications, had arrived.

THE COMPUTER GENERATIONS

“Generation” in computer took a step in technology. It provides a framework for the growth of the computer industry. Originally, the term “**generation**” was used to distinguish between varying hardware technologies. However, nowadays; it has been extended to include both hardware and software, which together make up an entire computer system.

The custom of referring to the computer in terms of generations came in to wide use only after 1964. There are totally five computer generations known until today. Each generation has been discussed below in detail along with its identifying characteristics. Although there is a certain amount of overlapping between the generations, the approximate dates shown against each are normally accepted.

1. FIRST GENERATION (1940-1956)

We have already discussed about some of the early computers – ENIAC, EDVAC, EDSAC, UNIVAC I. and IBM 701. These machines and others of their time were built by using thousands of vacuum tubes. A vacuum tube was a fragile glass device, which used filaments as a source of electronics, and could control and amplify electronic signals. It was the only high-speed electronic switching device available in those days. These vacuum tube computers could perform computations in milliseconds, and were referred to as first generation computers.

The memory of these computers were constructed using electromagnetic relays, and all data and instructions were fed in to the system from punched cards. The instructions were written in machine and assembly languages because high level programming languages were introduced much later. Because machine and assembly languages are very difficult to work with, only a few specialists understood how to program these early computers.

The characteristic features of first-generation computers are as follows:

1. They were the fastest calculating devices of their time.
2. They were too bulky in size, requiring large rooms for installation.
3. Thousands of vacuum tubes, which were used, emitted large amount of heat and burnt out frequently. Hence, the rooms/areas in which these computers were located had to be properly air-conditioned.
4. Each vacuum tube consumed about half a watt of power. Since a computer typically used more than ten thousand vacuum tubes, the power consumption of these computers was very high.
5. As vacuum tubes used filaments, they had a limited life. Since thousands of vacuum tubes were used in making one computer, these computers were prone to frequent hardware failures.
6. Due to low mean time between failures, these computers required almost constant maintenance.
7. In these computers, thousands of individual components had to be assembled manually by hand in to functioning circuits. Hence, commercial production of these computers was difficult and costly.
8. Since these computers were difficult to program and use, they had limited commercial use.

2. SECOND GENERATION (1956-1963)

A new electronic switching device, called *transistor* was invented at bell Laboratories in 1947 by John Bardeen, William Shockley, and Walter Brattain. Transistors soon proved to be a better electronic switching device than the vacuum tubes, due to their following properties:

1. They were more rugged and easier to handle than tubes, since they were made of germanium semiconductor material rather than glass.
2. They were highly reliable as compared to tubes, since they had no part like a filament, which could burn out.
3. They could switch much faster (almost ten times faster) than tubes. Hence, switching circuits made of transistors could operate much faster than their counterparts made of tubes.
4. They consumed almost one-tenth the power consumed by a tube.
5. They were much smaller than a tube.
6. They were less expensive to produce.
7. They dissipated much less heat as compared to vacuum tubes.

The second generation computers were manufactured using transistors, instead of vacuum tubes. Due to the properties of transistors listed above, the second generation computers were more powerful, more reliable, less expensive, smaller, and cooler to operate than the first generation computers.

The memory of the second-generation computers was composed of magnetic cores. Magnetic disk and magnetic tape were the main secondary storage media used in second generation computers. Punched cards were still popular and widely used for preparing programs and data to be fed to these computers.

The characteristics features of second generation computers are as follows:

1. They were more than ten times faster than the first generation computers.
2. They were much smaller than first generation computers, requiring smaller space.
3. Although the heat dissipation was much less than first generation computers the rooms/areas in which the second generation computers were located had to be properly air-conditioned.
4. They consumed much less power than the first generation computers.
5. They were more reliable and less prone to hardware failures than the first generation computers.
6. They had faster and larger primary and secondary storage as compared to first generation computers
7. They were much easier to program and use than the first generation computers. Hence, they had wider commercial use.
8. In these computers, thousands of individual transistors had to be assembled manually by hand in to functioning circuits. Hence, commercial production of these computers was difficult and costly.

3. THIRD GENERATION (1964-1971)

Although transistors were great deal of improvement over the vacuum tubes, they generated the heat and damaged the areas of the computer.

The Integrated Circuit(IC) was invented in 195 by jack Kilby .It combined electronic components on to a small silicon disc, made from quartz .more advancement made possible the fitting of even more components on a small chip or a semi conductor. Also in third generation computers, the operating system allowed the machines to

run many different applications. These applications were monitored and coordinated by the computers memory.

- 1959 – Jack Wilby of Texas instrumentation patented the first integrated circuit in Feb,1959,Kilby had made his first germanium IC in Oct,1958.Robert Noycee at Fairchild used planer process to make connections of components within a silicon IC in early 1959. The first commercial product using IC was hearing aid in Dec, 1963.General instrument made LSI chip (100+ components) for hammed organ 1968..

- 1964 – IBM product SABRE ,The first airline reservation tracking system for American airlines, IBM announced the system/360 all purpose computer ,using 8 bit character word length(a “Byte”) that was pioneered in the 7030 of April 1961 that grew out of the AF contract of Oct.1958 following Sputnik to develop transistor computers for BMEWS.

- 1968 – DEC introduced the first “mini computer”, the PDP-8, named after the mini skirt, Dec was founded in1957 by Kenneth H Olsen who came for the SAGE project at MIT and began sales of the PDP-1 in 1960.

- 1969 – Development began on ARPA net, founded by the DOD.

- 1971 – Intel produced large scale integrated (LSI) circuits that were used in the digital delay line, the first digital audio device.

4. FOURTH GENERATION COMPUTERS: (1971-PRESENT)

Fourth Generation Computers are the modern day computers .The size started to go down with the improvement in the integrated circuits. Very Large Scale (VLSI) and Ultra Large Scale (ULSI) ensured that millions of components could be fit in to a small chip. It reduced the size and price of the computers at the same time increasing the power, efficiency and reliability.” The Intel 4004 chip developed in 1971, took the integrated circuit one step further by locating all the components of a computer (central processing unit, memory and input, output controls) on a minuscule chip”.

Due to the reduction of cost and the availability of the computers power of a small place allowed everyday user to benefit. First, came the mini computers, which offered users different applications, most famous of these the word processors and spreadsheet which could be used by non

technical users. **Video game** system like Atari 2600 generated the interest of general populace in the computers.

In 1981, IBM introduced personal computers for home and office use. “The number of personal computers in use more than doubled from two million in 1981 to 5.5 million in 1982. Ten years later, 65 million PCs were used “Computer size kept getting reduced during the years. It went down from Desktop to laptops to Palmtops. Macintosh introduced Graphical user interface in which the user didn’t have to type instructions, but could use mouse for the purpose.

The continued improvement allowed the networking of computers for the sharing of data. Local area Network (LAN) and Wide Area Network (WAN), were potential benefits, in that they could be implemented in corporations and everybody could share data over it. Soon the internet and world wide appeared on the computer scene and formatted the high Tech revolutions of 90’s.

- 1971 – Hyatt at micro computer co. patented the microprocessor. Ted Hoff at intel in Feb introduced the 4 bit 4004, a VLSI of 2300 components, for the Japanese company Busicom to create a single chip for a calculator, IBM introduced the first 8-inch “memory disk”, as it was called then, or the “floppy disk” later. Hoffmann la roche patented the passive LCD display for calculators and watches, In Nov Intel announced the first micro computer, the MCS-4, Nolan

Bushnell designed the first commercial arcade video game “Computer space”.

- 1972 – Intel made the 8-bit 8008 and 8080 microprocessors, Gary Kildall wrote his control program or microprocessor (CP/M) disk operating system to provide instructions for floppy disk drives to work with the 8080 processor. He offered it to Intel, but was turned down. So he sold it on his own, and soon CP/M was the standard OS for 8 bit microcomputer.

5. FIFTH GENERATION COMPUTERS (PRESENT AND BEYOND)

First, Generation of computers is only in the minds of advance research scientist and being tested out in the laboratories. These computers will be under Artificial Intelligence (AI). They will be able to take commands in an audio visual way and carry out instructions. Many of

the operations which require low human intelligence will be performed by these computers.

Parallel processing is coming and showing the possibility that the power of many CPU's can be used side by side, and computers will be more powerful than those under central processing. Advances in super conductor technology will greatly improve the speed of information traffic. Futures look bright for the computers.

- 1991 – WWW (World Wide Web) was developed by Tim Berners-Lee and released by CERN.

- 1993 – The first Web Browser called Mosaic was created by student Marc Andreessen and programmer Eric Bina at NCSA in the first three months of 1993.

- 1994 – Netscape Navigator 1.0 was released 1994.

- 1996 - Microsoft failed to recognize the importance of the Web. But finally released the much improved Browser explorer.

The control unit and the arithmetic logic unit of a computer system are jointly known as the Central processing unit (CPU).The CPU is the brain of a computer system. In a human body, all major decisions are taken by the brain, and the other parts of the body function as directed by the brain. Similarly, in a computer system, all major calculations and comparisons are made inside the CPU, and the CPU is responsible for activating and controlling the operations of other units of the computer system.

ENIAC - 1946

- Electronic Numerical Integrator and Computer

- Under the leadership of *J. Presper Eckert* (1919 - 1995) and *John W. Mauchly* (1907 - 1980) the team produced a machine that computed at speeds 1,000 times faster than the Mark I was capable of only 2 years earlier.

- Using 18,000-19,000 vacuum tubes, 70,000 resistors and 5 million soldered joints this massive instrument required the output of a small power station to operate it.

- It could do nuclear physics calculations (in two hours) which it would have taken 100 engineers a year to do by hand.

- The system's program could be changed by rewiring a panel



Components Used In I Generation

- Electronic circuit used vacuum tubes.
- Punched cards used for feeding information.
- Punched cards and papers used for getting results.
- Magnetic tapes were used for external storage.
- Oriented towards batch processing, where the entire machine is dedicated to a particular job until completed.
 - Human operators had to set switches.
 - Development of machine language and assembly language.
 - Typical computers: UNIVAC-I, IBM 650, BURROUGHS 220.

Components used in II Generation

- Electronic circuit characterized by the use of transistors.
- Punched cards used for feeding information
- Punched cards and papers used for getting results.
- Magnetic core storage, i.e. magnetic tape used for external storage.
 - Human operators handled punched cards.
 - Development of high languages FORTRAN, COBOL, BASIC, PL/1 and others.

- Typical computers: IBM 1400 series and 7000 series, control data 3600.

Components used in III Generation

- Electronic circuit characterized by micro-electronic or integrated circuits which though microscopic in size, contained the equivalent of many transistors.
 - Monitors and keyboards were introduced for data input and output.
 - Punched cards began losing their prominence as the input or output device.
 - Magnetic disks used for external storage.
 - Designed around sophisticated operating systems capable of handling several jobs concurrently.
 - More high-level language including RPG (report Program Generator) and PASCAL were developed. Heavy emphasis was laid on applications-oriented languages-FORTRAN and COBOL.
 - Typical computer: IBM System/360.

Components used in IV Generation

The significant distinction for the fourth generation is the development of large-scale Integration (LSI) and Very Large Scale Integration (VLSI). LSI placed several thousand transistors onto a single chip. VLSI placed several hundred thousand transistors onto a single chip.

- Further miniaturization in which complete circuit are reduced to virtually microscopic sizes. Hundreds of circuits may be placed on a chip of the size of pinhead.
 - Further refinement of input and output devices.
 - Introduction of microcomputers.
 - Magnetic disk became the primary means of external storage.
 - The beginning of broad use of special software for managing large data bases.
 - Application software for microcomputers became popular.
 - Typical computers: IBM System/370, BURROUGHS B7700 and HP3000.

Components used in V Generation (Future)

The future may hold the following possibilities.

Superconductors advanced in artificial intelligence would be capable of billions of calculations per second. Computers that can ‘think’ and ‘reason’ cause further miniaturization of computers hardware. Artificial intelligence (AI) may be described as a branch of computer Science that is involved with using computers to solve problems that appear to require human imagination or intelligence. The public first became aware of this discipline in 1956 when the term was coined as the theme for a conference held at Dartmouth College. Since then, researchers have used concepts from disciplines such as linguistics, Psychology and Computer Science in an attempt to learn how to prepare programs or construct systems that do tasks which no machine has ever automatically done before.

An example of AI in action is the use of expert systems. An expert system is a software package that attempts to encode the knowledge and decision rules established by human specialists so that package users can call on this expertise in making their own decisions.

For the foreseeable future, the computers will continue to be intelligent amplifiers in an alliance with humanity. This alliance would combine the current superiority of the human brain in matters requiring processing -, speed, accuracy and tireless attention to detail.

INTRODUCTION

What Is a Computer?

A **computer** is an electronic device, operating under the control of instructions (software) stored in its own memory unit, that can accept data (input), manipulate data (process), and produce information (output) from the processing. Generally, the term is used to describe a collection of devices that function together as a system.

Characteristics of Computer

Computer has the following characteristics:

1. Speed
2. Accuracy
3. Reliability
4. Memory
5. Diligence
6. Automation
7. Versatile

Speed

A computer can add and subtract numbers, compare letters to determine alphabetic sequence, move and copy numbers and letters. As such, there is nothing profound in these operations. What is significant is the speed with which computers carry out these operations.

This speed varies from a few microseconds (millionth of a second) to nanoseconds (billionth of a second). For instance, a small computer can evaluate investment decisions by performing hundreds of thousands of machine operations in a second.

Accuracy

Computers are very accurate. They can perform their hundreds of thousands of operations with great accuracy as their circuits have no mechanical parts to wear and malfunction. They can run errorless for days at a time.

Reliability

Computer output is generally very reliable, subject to the condition that the input data entering the computer should be correct and the program of instructions should be reliable and correct. Incorrect input data and unreliable programs give us computer errors and wrong results. Hence, the phrase GARBAGE IN GARBAGE OUT (GIGO) .

Memory

Storage unit:

The process of saving data and instructions (commands) permanently is known as

Storage (secondary, primary storage)

Primary storage:

The primary storage, also known as main memory is used to hold pieces of program instructions and data intermediate results of processing and recently produced results of processing. The primary storage of modern computer systems is made up of semiconductor devices.

Secondary storage:

Secondary storage devices are the key for the data storage. They store the data for which the user wants to get back these data for future use. The examples for various secondary devices are Floppy disk, Optical disks (CD and DVD), Zip drives, Thumb drives etc. The data of smaller size can be easily fetched and they can be copied to the primary memory (RAM).

Diligence

A computer can work tirelessly for a long time. The amount of work that is given to the computer does not tire it out.

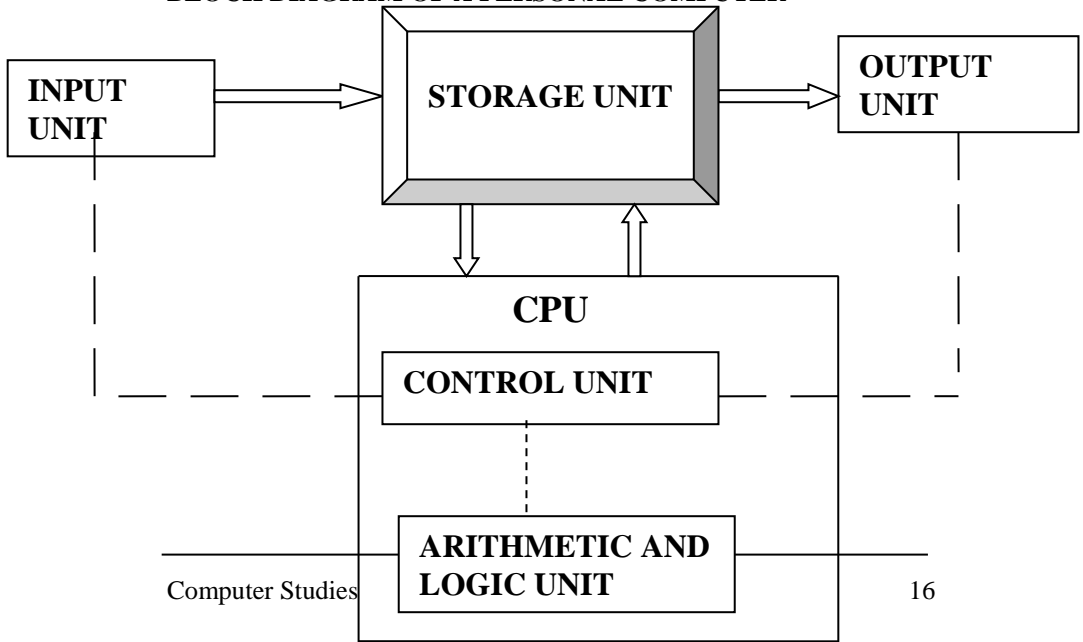
Automation

The computers are automatic. It may execute the process without any intervention of user once they are assigned to a work. Once the data or instruction are fetched from the secondary devices such as optical disks, hard disks etc. Immediately they get stored into RAM (primary memory) and then sequentially they get executed.

Versatile

In our day today life computers has been a part, with their extended flexibility they are used, all over the world. They can be used as personal computers, for home uses, for business oriented tasks, weather forecasting, teaching, railways, banking, medicine etc. All Modern computers can perform different kind of tasks simultaneously.

BLOCK DIAGRAM OF A PERSONAL COMPUTER



Input unit

Data and instructions must enter the computer system, before any computation can be performed other supplied data. This task is performed by the input unit, which links the external environment with the computer system. Data and instructions enter input units in forms, which depend upon the particular device used.

The following functions are performed by an input unit:

1. It accepts (or reads) the instructions and data from the outside world.
2. It converts these instructions and data in computer acceptable form.
3. It supplies the converted instructions and data to the computer system for further processing.

Input devices are: Keyboard, Scanner, Digital Camera etc.

Output unit

The job of an output unit is just the reverse of that of an input Unit., It supplies the information obtained from data processing, to the outside world. Hence it links the computer with the external environment's computers work with binary code; the results produced are also in the binary form. Hence, before supplying the results to the outside world, they must be converted to human acceptable (readable) form. This task is accomplished by units called output interfaces. Output interfaces are designed to match the unique physical or electrical characteristics of output devices to the requirements of the external environment.

The following functions are performed by an output unit:

1. It accepts the results produced by the computer, which are in coded form, and hence cannot be easily understood by us.
2. It converts these coded results to human acceptable form
3. It supplies the converted results to the outside world

Output devices are: Monitor, Printers, and Speakers.

Storage unit:

The process of saving data and instructions (commands) permanently is known as storage (secondary, primary storage). Storage Device: Hard drive, Floppy drive, CD drive.

Primary storage:

The primary storage, also known as main memory is used to hold pieces of program instructions and data intermediate results of processing and recently produced results of processing. The primary

storage of modern computer systems is made up of semiconductor devices.

Secondary storage:

The secondary storage also known as auxiliary storage is used to take care of the limitations of the primary storage. That is it is used to supplement the limited storage capacity and the volatile characteristic of primary storage. The most commonly used secondary storage medium is the magnetic disk.

Central processing unit

The control unit and the arithmetic logic unit of a computer system are jointly known as the Central processing unit (CPU).The CPU is the brain of a computer system. In a human body, all major decisions are taken by the brain, and the other parts of the body function as directed by the brain. Similarly, in a computer system, all major calculations and comparisons are made inside the CPU, and the CPU is responsible for activating and controlling the operations of other units of the computer system.

CPU Operation:

The fundamental operation of most CPUs

- To execute a sequence of stored instructions called a program.

- The program is represented by a series of numbers that are kept in some kind of computer memory.
- There are four steps that nearly all CPUs use in their operation: fetch, decode, execute, and writeback.

Fetch:

- Retrieving an instruction from program memory.
- The location in program memory is determined by a program counter (PC)
- After an instruction is fetched, the PC is incremented by the length of the instruction word in terms of memory units.

Decode :

- The instruction is broken up into parts that have significance to other portions of the CPU.
- The way in which the numerical instruction value is interpreted is defined by the CPU's instruction set architecture (ISA).
- Opcode, indicates which operation to perform.

- The remaining parts of the number usually provide information required for that instruction, such as operands for an addition operation.
- Such operands may be given as a constant value or as a place to locate a value: a register or a memory address, as determined by some addressing mode.

Execute :

- During this step, various portions of the CPU are connected so they can perform the desired operation.
- If, for instance, an addition operation was requested, an arithmetic logic unit (ALU) will be connected to a set of inputs and a set of outputs.
- The inputs provide the numbers to be added, and the outputs will contain the final sum.
- If the addition operation produces a result too large for the CPU to handle, an arithmetic overflow flag in a flags register may also be set.

Write back :

- Simply "writes back" the results of the execute step to some form of memory.
- Very often the results are written to some internal CPU register for quick access by subsequent instructions.
- In other cases results may be written to slower, but cheaper and larger, main memory.
- Some types of instructions manipulate the program counter rather than directly produce result data.

Arithmetic logic unit

The arithmetic logic unit (ALU) of a computer system is the place, where the actual execution of the instructions takes place, during the processing operation. To be more precise, calculations are performed, and all comparisons are made in the ALU. The data and instructions, stored in the primary storage before processing are transferred as and when needed to the ALU, where processing takes place. No processing is done in the primary storage unit.

Control unit

How does the input device know that it is time for it to feed data in to the storage unit? How does the ALU know, what should be done with the data once they are received? Moreover, how is it that only the final results are sent to the output device, and not the intermediate results?

All this is possible due to the control unit of the computer system. Although, it does not perform any actual processing on the data, the control unit acts as a central nervous system, for the other components of the computer system.

Memory

Bit: The smallest piece of information used by the computer. It is derived from a "binary digit". Which is either a one (1) or a zero (0).

Byte: A piece of computer information made up of eight bits.

- **WORD: 4 bytes = 32 bits**
- **Byte – 8 Bits**
- **Kilo Byte – 1024 bytes**
- **Mega Byte – 1024 Kilo Bytes**
- **Giga Byte – 1024 Mega Bytes**
- **1 Tera Byte – 1024 Giga Bytes**

SEMICONDUCTOR MEMORY:

Semiconductor memory consists of integrated electronic circuit attached into silicon chips. Smaller in size, higher range processing speed, use of less power and cheaper cost and easy to assemble and to maintain.

It has been classified as two types

- RAM
- ROM

RAM:

- RAM is acronym for Random-Access Memory.
- Main Memory.
- Temporary-It is a volatile memory, i.e. it cannot hold data when power is turned off. It is often called Read/Write memory because the data can be read from or writes into the RAM.
- It is where programs and data is stored while being processed

Types of RAM:

1. **DRAM:** Acronym for **Dynamic Random Access Memory**.It must be refreshed frequently. The term Refreshing means recharging the RAM chips with electricity.

DRAM chips must be recharged many times each second, or they will lose their contents.

Standard DRAM supports data transfer rates of about 60 nano seconds. This seems very fast, and it is, but other types of RAM are actually much faster than standard DRAM.

2. **SRAM:** Acronym for **Static Random Access Memory**. SRAM does not need to be refreshed as often as DRAM and can hold its contents longer. SRAM is considerably faster than DRAM, but it is also more expensive. For this reason, SRAM is used only for special purposes and not as a computer's primary memory.
3. **SDRAM:** Acronym for **Synchronous Dynamic Random Access Memory**. SDRAM operates faster than standard DRAM because it runs "in sync" with the computer's system clock. By running at the same speed, the CPU and SDRAM can move data much more efficiently than is possible with DRAM.
4. **RDRAM-** Acronym for **RAMBUS DYNAMIC RAM & DDRSDRAM--** Acronym for **DOUBLE DATA RATE SDRAM**. These are both newer and faster variations on DRAM, and are commonly used in today's newer computer systems.

ROM:

- ROM is acronym for Read Only Memory.
- Auxiliary memory.
- Instructions are installed permanently at the factory
- These instructions check the computer's resources and look for Operating System

Different types of ROM

1. **PROM-** Acronym for **Programmable Read –Only Memory**. This is a non-volatile (stable) storage. The user can program it. PROM's are programmed to record information using a device called PROM programmer. Once it is programmed it becomes a ROM. We can't change the programme.
2. **EPROM-** Acronym for **Erasable Programmable Read only Memory**. It is possible to erase information stored in an EPROM chip, and the chip can be reprogrammed to store new information. When an EPROM is in use, information stored in it can only be "read" and the information remains in the chip, until it is erased. UVEPROM (Ultraviolet Erasable Programmable Read only Memory).

3. **EEPROM-** Acronym for **Electrically Erasable Programmable Read only Memory**. It is an electronically erasable and programmable read-only memory chip that can be reprogrammed with special electric pulse. Is a type of EPROM where data is erased using electricity.

Floppy Disks

Floppy disks are removable storage media. Today's standard is 1.44 MB. Data recorded in rings called tracks; tracks divided into wedge-shaped sections called sectors. Most disks manufactured currently preformatted w/tracks and sectors while called floppy, actually pieces of flat Mylar rotating within a hard plastic jacket. The bits of data are represented by the presence or absence of magnetic charges

It is a small plastic disk enclosed in a stiff envelop with a radial slit, used to store data or program.

Floppy disks basically come in two sizes- floppy disks:

1. 5.25"



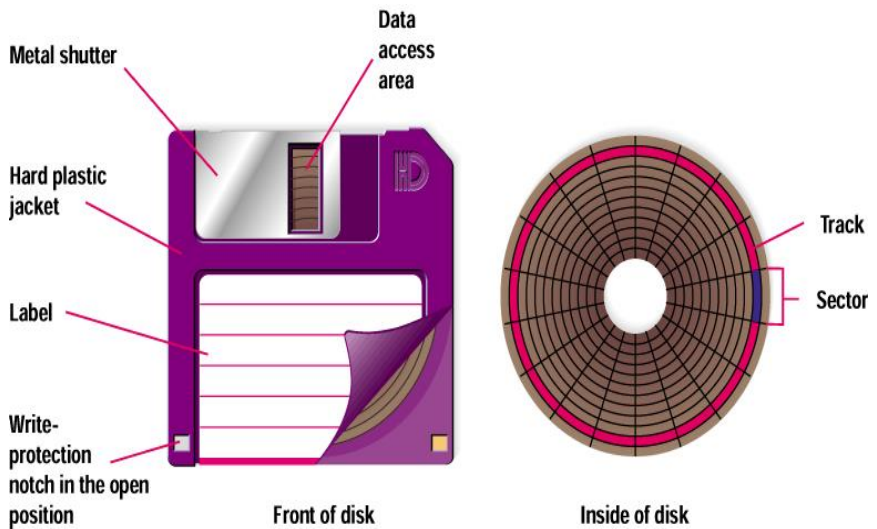
2. 3.5"



Floppy disks have three common features:

1. Label area
2. Read/ write opening
3. Write protect notch

Traditional Diskette



1. Label area:

This area has been assigned to the label that can be fixed on the floppy which indicates what information/data is stored in it.

2. Read/write opening:

Through this area, the computer is able to read the contents from the floppy or write the contents to the floppy.

It is advisable never to touch this area as this may lead to damage of data stored in the floppy.

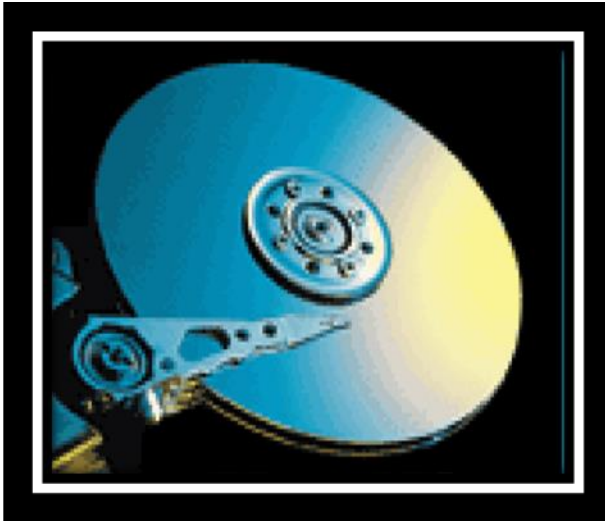
3. Write Protect Notch:

This prevents you or anyone else from accidentally or deliberately changing or erasing anything from the diskette.

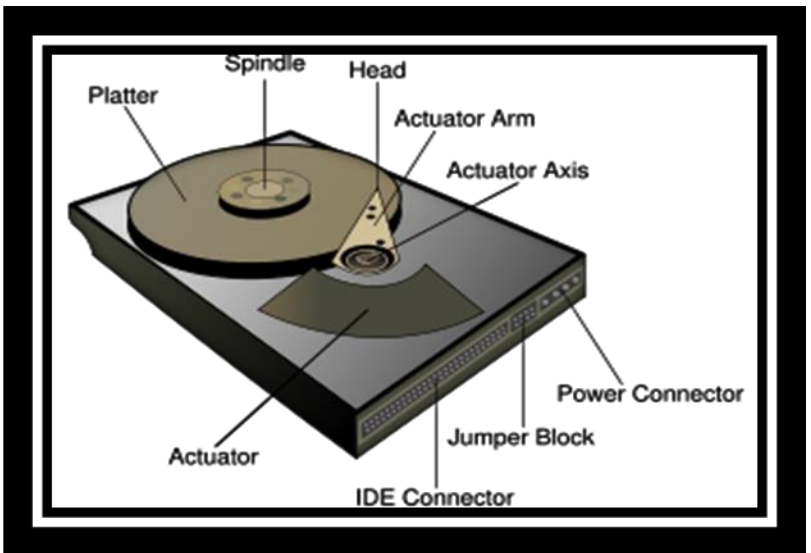
It means when the diskette is **write protected**.

HARD DISK COMPONENTS

1. Disk Platter
2. Read / Write Head
3. Head Arm / Head Slider
4. Head Actuator Mechanism
5. Spindle Motor
6. Logic Board
7. Air Filter
8. Cables & Connectors



- **Form Factor**
 - 5.5" – actual size is 5.12"
 - 3.5" – actual size is 3.74"
 - 2.5"
 - 1 1/8"
 - 1 1/3"
 - 1"



SPINDLE MOTOR

- It is the main motor which rotates the hard disk drive platter. It is called spindle motor because it is directly connected to the spindle on which the platters are connected. Spindle motor rotates at a speed of 3600 to 7200 rpm or more.

LOGIC BOARD

- An intelligent circuit board is built into the hard disk in the modern days. It contains the electronic components that control various sections of the hard disk drive. It also acts as an interface between the hard disk drive and the computer.

AIR FILTER

- To filter the air. Most hard disk drives have two air filters:
 - Recirculating Air Filter
 - Breather Filter

HEAD ACTUATOR MECHANISM

- The Read/Write head is moved on the platter using a mechanism
- Two types of mechanism :
 - Stepper Motor Actuator
 - Voice Coil Actuator

STEPPER MOTOR ACTUATOR

- Rotates in steps.
- Turns in a fixed angle
- The smallest fixed angle is called as “Detent”
- Connected with Read / Write head by
 - Split Metal Band Mechanism
 - Rack & Pinion Gear Mechanism

SIDES & HEADS

- Each platter has two sides
- Each side has each head
- Total no. of sides = Total no. of heads
- When addressing either, heads or sides are denoted
- Total no. of heads may range from 2 – 256
- Total no. of sides may range from 2 – 20
- The head numbering starts from 0 . . .
- It starts from the uppermost of the platter

TRACKS

- Each side of the platter's surface is divided in to so many concentric circles
- The track numbering starts from 0 . . .
- The outermost track is given the track no. 0
- The innermost track will have the highest no.
- No. of tracks on a platter may range from 300 - 3000
- Modern hard disks have thousands of tracks on each platter

CYLINDERS

- Some tracks on different platters or sides form an imaginary cylinder
- The data is stored in a cylinder by cylinder method due to single head assembly
- A track and cylinder are two different things, but they are used simultaneously
- Total no. of tracks = Total no. of cylinders
- The cylinder numbering starts from 0..., It starts from the outermost of the platter

SECTORS

- It is the smallest unit of data storage
- It can hold up to 512 MB of data
- The sector numbering starts from 1...
- The storage capacity of the hard disk drive varies according to the number of sectors / tracks present in it.
- A group of sectors is called as "Cluster"
- It is the smallest unit of data storage which can be identified by the operating system

Peripheral Devices

1. Monitor
2. Keyboard
3. Central Processing Unit (CPU)
4. Mouse

Input devices:

Keyboard:

The keyboard was one of the first peripherals to be used with computers, and it is still the primary input device for entering text and

numbers. A standard keyboard includes about 101 keys; each key sends a different signal to the CPU.

If you have not used a computer keyboard or a typewriter, you will learn quickly that you can use a computer much more effectively if you know how to type. The skill of typing or keyboarding is the ability to enter text and numbers with skill and accuracy. Certainly, you can use a computer without having good typing skills.

The Standard keyboard layout:

Keyboards for personal computers come in many styles. The various models differ in size, shape, and feel, but except for a few special-purpose keys, most keyboards are laid out almost identically. Among IBM-compatible computers, the most common keyboard layout is the IBM Enhanced keyboard. It has 101 keys arranged in five groups,

- The Alphanumeric keys
- The Modifier keys
- The function keys
- The cursor movement keys
- Special purpose keys.

The Alphanumeric keys:

The alphanumeric keys-are the parts of the keyboard that look like a typewriter arranged in the same way on almost every keyboard. Sometimes this common arrangement is called the QWERTY(Pronounced KWER-tee) layout because the first six keys on the top row of letters are Q, W, E, R, T, and Y. Along with the keys that produce letters and numbers, the alphanumeric key group includes four keys that have specific functions. The Tab, Caps Lock, Backspace and Enter keys.

The Modifier keys:

A keyboard's modifier keys are so named because they modify the input of other keys. In other words, if you hold down a modifier key while pressing another key, then you are changing the keys input in some way. Modifier keys are extremely useful, because they give all other keys multiple capabilities. Modifier keys are shift, ctrl, alt.

The function keys:

The fourth part of the keyboard consists of the function keys. These keys (labeled F1, F2, and so on) are usually arranged in a row along the top of the keyboard. They allow you to input commands without typing long strings of characters or navigating menus or dialog boxes. Each function Key's purpose depends on the program you are using, For example, in most programs, F1 is the help key.

The cursor movement keys:

The fifth part of the keyboard is the set of cursor movement keys, which let you move around the screen without using a mouse. In many programs and operating systems, a mark on the screen indicates where the characters you type will be entered. This mark, called the cursor or insertion point, appears on the screen as a blinking vertical line, a small box, or some other symbol to show your place in a document or command line.

Special purpose keys:

Start, This key, which features the windows logo, opens the windows start menu on most computers. Processing this key performs the same function as clicking the start button on the windows taskbar.

Shortcut, This key, which features an image of a menu, opens an on-screen shortcut menu in windows based application programs.

**Mouse:**

A mouse is an input device that rolls around on a flat surface and controls the pointer. The pointer (also called the mouse pointer) is an on-screen object-usually an arrow that is used to select text, access menus, and interact with programs, files or data that appear on the screen.

Most mice are mechanical; they contain a small rubber ball that protrudes through a hole in the bottom of the mouse's case. The ball rolls inside the case when you move the mouse around on a flat surface. Inside the mouse, rollers and sensors send signals to the computer

telling it the distance, direction, and speed of the balls motions. The computer uses this data to position the mouse pointer on the screen.

OCR:

OCR represents Optical Character Recognition Device

When image scanners are used for inputting text documents (typed or handwritten), they have the following two limitations;

1. Since the input documents are stored as an image, instead of text, it is not possible to do any word processing of the document.
2. The storage required for storing the document as an image is much more than that required for storing the same document as a text.

The OCR technology is used to overcome these limitations. In this case, the scanner is equipped with a character recognition software (called OCR software), which converts the bit map images of characters to equivalent ASCII codes. That is, the scanner first creates the bit map image of the document, and then the OCR software translates the array of grid points in to ASCII text, which the computer can interpret as letters, numbers and special characters.

OCR software is extremely complex, because it is difficult to make a computer recognize an unlimited number of typefaces and fonts. Hence, these software of characters to equivalent ASCII codes.

MICR:

MICR represents **Magnetic –ink Character Recognition**. Its similar to OCR, and is used by the banking industry for faster processing of the large volume of cheques being handled every day by this industry. Banks which Emory MICR technology, use a special type of cheque. The bank's identification code (name, branch, etc.), account number, and the cheque number are pre-printed(encoded) by using characters from a special character set on all these cheques with a special ink, which contains magnetizable particles of iron oxide, before the cheques are given to the customers for use.

When a customer presents a filled –in cheque at a bank, a bank employee manually enters(keys in)the amount written on the cheque, In the lower right corner of the cheque by using an MICR inscriber, which prints the amount with the magnetic ink. The date of the transaction is automatically recorded for all cheques processed that day.

This cheque is then processed by using an MICR reader-sorter, which can recognize magnetic ink characters. The MICR reader-sorter reads the data on the cheques, and sorts the cheques for distribution to other banks, or for further processing.

The most commonly used character set by MICR devices is known as E13B font, which consists of the numerals 0 to 9, and four special characters. Coded data in the form of these fonts are transferred from cheques to the computer by an MICR reader sorter. As the cheques enter the reading unit, they pass through a magnetic field, which cause the particles in the ink to become magnetized. Read heads then interpret these characters by examining their shapes. The sorter is used to sort the cheques in to different packets, according to their identification code numbers.

MICR technology speeds up data input for the banking industry, because cheques can be directly fed into the input device, and automatically processed thereafter. Besides enabling faster processing of cheques, this technology also ensures accuracy of data entry, because most of the information is preprinted on the cheque, and is directly fed in to the computer.

Output devices:

VDU:

VDU stands for **Visual Display Unit** Monitors are the most popular output device used for producing soft-copy output. They display the generated output on a television like screen. A monitor is usually associated with a keyboard, and together they form a video display terminal (VDT). A VDT is the most popular I/O device used with today's computers. That is, it serves as both an input and an output device. The keyboard is used for input to the computer, and the monitor is used to display the output from the computer. The name "terminal" comes from the fact that a terminal is at the terminal, or end, point of a communication path.

The two basic types of monitors used today are cathode-ray-tube (CRT) and flat-panel. The CRT

Monitors look much like a television, and are used with non portable computer systems. On the other hand, the flat panel monitors are thinner and lighter, and are commonly used with portable computer systems, like notebook computers.



Printers

Printers are hardware devices that allow you to create a hard copy of a file. Today a printer is a necessary requirement for any home user and business. Allowing individuals to save their work in the format of paper instead of electronically.

Types of Printers

- **Impact printers**

In case of Impact printer an inked ribbon exists between the print head and paper, the head striking the ribbon prints the character.

- **Non Impact Printers**

Non Impact printers use techniques other than the mechanical method of head striking the ribbon.

Impact printers

Impact printers are basically divided into 2 types

- Serial/Character printers
- Dot matrix printers

- Daisy wheel printers
- Line Printers

Non-Impact Printers

- Non Impact Printers are divided into 3 categories
- Thermal printers
- Ink jet printers
- Laser printers

SERIAL /CHARACTER PRINTER:

A printer that prints one character at a time in the sequence in which they appear in the line of text. The sequence may be taken from left to right, or it may be in alternate directions for alternate lines thus avoiding an unproductive carriage-return movement. All serial printers have an arrangement in which a print head moves parallel to the paper and along the line to be printed.

DOT MATRIX PRINTERS:

Dot-matrix printers are character printers, which print one character at a time. They form characters and all kinds of images as a pattern of dots. A dot-matrix printer has a print head, which can move horizontally (left to right and right to left) across the paper. The print head contains an array of pins, which can be activated independent of each other to extend strike against an inked ribbon to form a pattern of dots on the paper. To print a character, the printer activates the appropriate set of pins as the print head moves horizontally. For faster printing, many dot matrix printers are designed to print both while the print head moves from left to right, and while it moves from right to left, on return.

Since dot matrix printers produce printed output as patterns of dots, they can print any shape of character, which a programmer can describe. This allows the printer to print many special characters, different sizes of print, and the ability to print graphics such as charts and graphs.

Dot-matrix printers are impact printers, because they print by hammering the pins on the inked ribbon to leave ink impression on the paper. Hence, they can be used to produce multiple copies by using carbon paper or its equivalent. Dot-matrix printers are noisy as compared to non-impact printers.

Dot-matrix printers are normally slow with speeds usually between 30 to 600 characters per second. However, they are cheap in terms of both initial cost and cost of operation. Hence they are preferred by individuals and organizations for generating printed outputs, if the speed and quality of printing are not important factors. They are also used for applications, which require multi copy output, such as shipping forms and invoices, which rely on impact for generating multiple copies

For instance, if you use multipart forms, dot matrix is the only technology that can handle the thickness of these documents and print through all the layers. It is also ideal for those who need reliable printers to put out large quantities of mailing labels, envelopes, reports, or packing labels. Dot matrix printers are generally a cost-effective solution for those who need big jobs done quickly.

Daisy Wheel Printer:

A daisy wheel printer is a type of computer printer that produces high-quality type, and is often referred to as a letter-quality printer (this in contrast to high-quality dot-matrix printers, capable of near-letter quality, or NLQ, output). There were also, and still are daisy wheel typewriters, based on the same principle. The DWP is slower the speed range is in 30 to 80 CPS.

The system used a small wheel with each letter printed on it in raised metal or plastic. The printer turns the wheel to line up the proper letter under a single pawl which then strikes the back of the letter and drives it into the paper. In many respects the daisy wheel is similar to a standard typewriter in the way it forms its letters on the page, differing only in the details of the mechanism (daisy wheel vs type bars or the type ball used on IBMs electric typewriters).

Daisy wheel printers were fairly common in the 1980s, but were always less popular than dot matrix printers (ballistic wire printers) due to the latter's ability to print graphics and different fonts. With the introduction of high quality laser printers and inkjet printers in the later 1980s daisy wheel systems quickly disappeared but for the small remaining typewriter market.

Line Printer:

The line printer is a form of high speed impact printer in which a line of type is printed at a time. The wheels spin at high speed and paper and an inked ribbon are stepped (moved) past the print position. As the

desired character for each column passes the print position, a hammer strikes the paper and ribbon causing the desired character to be recorded on the continuous paper. The speed is 300 to 2500 lines per minute (LPM). This technology is still in use in a number of applications. It is usually both faster and less expensive (in total ownership) than laser printers. In printing box labels, medium volume accounting and other large business applications, line printers remain in use

Line printers, as the name implies, print an entire line of text at a time. Two principle designs existed. In drum printers, a drum carries the entire character set of the printer repeated in each column that is to be printed. In chain printers (also known as train printers), the character set is arranged multiple times around a chain that travels horizontally pas the print line. In either case, to print a line, precisely timed hammers strike against the back of the paper at the exact moment that the correct character to be print edis passing in front of the paper. The paper presses forward against a ribbon which then presses against the character form and the impression of the character form is printed onto the paper.

These printers were the fastest of all impact printers and were used for bulk printing in large computer centers. They were virtually never used with personal computers and have now been partly replaced by high-speed laser printers.

Thermal Printers:

Direct thermal printers create an image by selectively heating coated paper when the paper passes over the thermal print head. The coating turns black in the areas where it is heated, creating the image. More recently, two-color direct thermal printers have been produced, which allow printing of both red (or another color) and black by heating to different temperatures.

Thermal Printer Characters are formed by heated elements being placed in contact with special heat sensitive paper forming darkened dots when the elements reach a critical temperature. A fax machine uses a thermal printer. Thermal printer paper tends to darken over time due to exposure to sunlight and heat. The standard of print produced is poor. Thermal printers are widely used in battery powered equipment such as portable calculators.

Direct thermal printers are increasingly replacing the dot matrix printer for printing cash register receipts, both because of the higher print speed and substantially quieter operation. In addition, direct thermal printing offers the advantage of having only one consumable - the paper itself. Thus, the technology is well-suited to unattended applications like gas pumps, information kiosks, and the like.

Until about 2000, most fax machines used direct thermal printing, though, now, only the cheapest model use it, the rest having switched to either thermal wax transfer, laser, or ink jet printing to allow plain paper printouts. Historically, direct thermal paper has suffered from such limitations as sensitivity to heat, abrasion (the coating can be fragile), friction (which can cause heat, thus darkening the paper), light (causing it to fade), and water. However, more modern thermal coating formulations have resulted in exceptional image stability, with text remaining legible for an estimated 50+ years.

INK-JET PRINTERS:

Inkjet printers are character printers, which form characters and all kinds of images by spraying small drops of ink on to the paper. The print head of an inkjet printer contains up to 64 tiny nozzles, which can be selectively heated up in a few microseconds by an integrated circuit register. When the register heats up, the ink near it vaporizes, and is ejected through the nozzle, and makes a dot on the paper placed in front of the print head. To print a character, the printer selectively heats the appropriate set of nozzles as the print head moves horizontally.

Inkjet printers produce higher quality output than dot-matrix printers, because they form characters by very tiny ink dots. A high-resolution inkjet printer has as many as 64 nozzles within height of 7mm, providing print resolution of around 360 dots per inch.

Since inkjet printers produce printed output as patterns of tiny dots, they can print any shape of characters, which a programmer can describe. This allows the printer to print many special characters, different sizes of print, and the ability to print graphics, such as charts and graphs.

Inkjet printers are non-impact printers, because they print by spraying ink on the paper. Hence, they are quiet in operation. However, like other non-impact printers, inkjet printers cannot be used to produce multiple copies of a document in a single printing.

Inkjet printers are slower than **dot-matrix printers** with speeds usually ranging between 40 to 300 characters per second. Typically, an inkjet printer is more expensive than a dot-matrix printer. They are preferred by individuals and organizations for generating printed outputs, if the speed of printing is not an important factor.

Although inkjets were available in the 1980s, it was only in the 1990s that prices dropped enough to bring the technology to the high street. Canon claims to have invented what it terms '*bubble jet*' technology in 1977, when a researcher accidentally touched an ink-filled syringe with a hot soldering iron. The heat forced a drop of ink out of the needle and so began the development of a new printing method.

Inkjets have one massive attraction over laser printers; they produce colour, and that is what makes them so popular with home users. The down side is that although inkjets are generally cheaper to buy than lasers, they are more expensive to maintain. Cartridges need to be changed more frequently and the special coated paper required to produce high-quality output is very expensive. When it comes to comparing the cost per page, inkjets work out about ten times more expensive than laser printers.

LASER PRINTERS:

Laser Printers are page printers, which print one page at a time. The main components of a laser printer are a laser beam source, a multi-sided mirror, a photoconductive drum and toner (tiny particles of oppositely charged ink). To print a page of output, the laser beam is focused on the electro statically charged drum by the spinning multisided mirror. The mirror focuses the laser beam on the surface of the drum in a manner to create the patterns of characters/images to be printed on the page. As the drum is photoconductive, a difference in electric charge is created on those parts of the drum surface, which are exposed to the laser beam. As a result, the toner, which is composed of oppositely charged ink particles, sticks to the drum in the places, where the laser beam has charged the drum's surface. The toner is then permanently fused on the paper with heat and pressure to generate the printed output. The drum is then rotated and cleaned with a rubber blade to remove the toner sticking to its surface to prepare the drum for the next page printing.

Laser printers produce very high quality output, because they form characters by very tiny ink particles. The most common laser printers have resolution of 600 dpi (dots per inch), whereas some high-end laser printers have resolution of 1200 dpi.

Laser printers are non-impact printers, because they do not have hammers striking on an inked ribbon or paper. Hence, they are very quiet in operation. However, like other non-impact printers, laser printers cannot be used to produce multiple copies of a document in a single printing.

Laser printers are faster in printing speed than other printers discussed before. Low speed laser printers can print 4 to 12 pages per minute. Very high-speed laser printers are also available, which can print 500 to 1000 pages per minute. Laser printers are more expensive than other printers.

PLOTTERS:

Everyone knows about dot matrix, inkjet and laser printers are capable of producing graphics output. However, many engineering design applications, like architectural plan of a building, design of mechanical components of an aircraft or a car, etc., often require high-quality, perfectly-proportioned graphic output on large sheets. Laser, inkjet, dot-matrix are not suitable for meeting this output requirement of such applications. A special type of output device, called plotters.

The two commonly used types of plotters are drum plotter and flatbed plotter.

Drum Plotter

In a drum plotter the paper, on which the design has to be made is placed over a drum, which can be rotate in both clockwise and anticlockwise directions to produce vertical motion. The mechanism also consists of one or more penholders mounted perpendicular to the drum's surface. The pen(s) clamped in the holders can move left to right or right to left to produce horizontal motion. The movements of the drum and the pens are controlled by the graph-plotting program. That is, under computer control, the drum and the pens move simultaneously to draw the designs and graphs on the sheet placed on the drum.

Flatbed plotter

A flatbed plotter plots a design or graph on a sheet of paper, which that is spread and fixed over a rectangular flatbed table. In this type of plotter normally the paper does not move, and the pen holding mechanism is designed to provide all types of motions necessary to draw complex designs and graphs. That is, under computer control, the pens move in the required manner to draw the designs and graphs on the sheet placed on the flatbed table.

CLASSIFICATION OF COMPUTERS

The computers are categorized in to the following types:

- **Notebook Computers**

Notebook computers are portable computers, which are mainly meant for use by people who computing need power wherever they go. As the name implies, notebook computers are approximately of the size of an 8(1/2) X 11 inch notebook, and can easily fit inside a briefcase. Since they have to be carried along, they are also light in weight, weighting around 2Kg.They are also known as laptop PCs (laptop personal computers), because they are as powerful as a PC, and their size and weight allows them to be used by comfortably placing them on ones lap.

A notebook computer uses an almost full-size keyboard, a small flat screen liquid crystal color display, and a trackball(instead of a mouse, because notebook computers are often used without a desk).They also have a hard disk, a floppy disk drive, and a CD-ROM drive. The display screen is foldable in a manner that, when not in use, it can be folded to flush with the keyboard, to convert the system in to notebook form. When in use, the display screen is folded open. Many models of notebook computers can be plugged into a “dock” on a docking station (a personal computer or a workstation), to take advantage of the larger machines big monitor, storage space, and other peripherals, such as a printer.

Many models of notebook computers can also be connected to a network, to enable them to download data (read files) from other computers on the network.

- **Personal Computers(PCS)**

Personal computers are desktop computers designed for an individual's use. They run programs designed to help individuals accomplish their work more productively.

A PC is a non-portable, general-purpose computer, which can easily fit on a normal size office table. PCs are found on the working desks of several employees, in any organizations. PCs are also providing employees with flexible working environments. Those employees, who could not work during traditional office hours due to personal reasons, can now work part of the time in the office and the remainder of the time at home, by having a PC in their homes. PCs are now very common everywhere, and can be found in offices, classrooms, homes, hospitals, shops, clinics, etc.

- **Workstations**

A workstation is a powerful desktop computer, which is designed to meet the computing needs of engineers. Architects, and other professionals, who need greater processing power, larger storage and better graphic display facility than what PCs provide.

A workstation looks very much like a PC, and is typically used by only one person at a time, just like a PC. The following characteristics are often used to differentiate between the two:

Classification of computer based on Computing power

Super Computers:

Super-computers are very fast and powerful machines. Their internal architecture enables them to run at the speed of tens of MIPS (Million Instructions per Second). Super-computers are very expensive and for this reason are generally not used for CAD applications. Examples of super-computers are: Cray and CDC Cyber 205.

Mainframe Systems:

A mainframe is the largest computer, a powerhouse with massive memory and extremely rapid processing power. It is used for very large commercial, scientific, or military applications where a computer must handle massive amounts of data or many complicated processes.

There are several organizations such as banks, insurance companies, hospitals, railways, etc. Mainframe systems are computer systems, that need on-line processing of large number of transactions, and require computer systems, which have massive data storage and processing capabilities.

A typical configuration of a mainframe system consists of the following components:

Host, front-end, and back-end computers.

Console(s).

Storage devices.

User terminals.

Output devices.

In early days of computing Mainframes are big computers that can fill the entire room or entire floor. Mainframes are mainly used by large organizations for critical applications, such as consumer statistics, ERP, and financial transaction processing.

Large organizations use mainframes for highly critical applications such as bulk data processing and ERP. Most of the mainframe computers have the capacities to host multiple operating systems and operate as a number of virtual machines and can thus substitute for several small servers.

A mainframe is the largest computer, a powerhouse with massive memory and extremely rapid processing power. It is used for very large commercial, scientific, or military applications where a computer must handle massive amounts of data or many complicated processes.

Generally, however, mainframes can be classified as having 50 megabytes to over 1 gigabyte of RAM.



Minicomputers

- Multi-user systems that can handle the computing needs of a smaller corporation or organization.
- Many people use them simultaneously by means of remote terminals or personal computers.
- The **mini computers came** which offered user's different applications, most famous of these the **word processors and spreadsheet** which could be used by non technical users. **Video game system** like Atari 2600 generated the interest of general populace in the computers.
- In terms of size and processing capacity, minicomputers lie in between mainframes and microcomputers. Minicomputers are also called mid-range systems or workstations. The term began to be popularly used in the 1960s to refer to relatively smaller third generation computers. They took up the space that would be needed for a refrigerator or two and used transistor and core memory technologies. The 12-bit PDP-8 minicomputer of the Digital Equipment Corporation was the first successful minicomputer.
- Minicomputers lie between the **Mainframe computer** and small **Personal Computers**. Minicomputers also referred to **Mid-Range Servers**. Minicomputers are more powerful but still compatible version of a personal computer.
- A minicomputer is a middle-range computer, about the size of an office desk, often used in universities, factories, or research laboratories.
- Minicomputers, 10 megabytes to over 1 gigabyte of RAM.



Micro Computers:

Microcomputers were invented in the 1970's and were generally used for home computing and dedicated data processing workstations.

Advances in technology have improved micro computer capabilities, resulting in the explosive growth of personal computers in industry. In the 1980's many medium and small design firms were finally introduced to CAD as a direct result of the low cost and availability of microcomputers. Examples are: IBM, Compaq, Dell, Gateway, and Apple Macintosh.

The average computer user today uses a microcomputer. These types of computers include PC's, laptops, notebooks, and hand-held computers such as Palm Pilots.

Larger computers fall into a mini-or mainframe category. A mini-computer is 3-25 times faster than amicro. It is physically larger and has a greater storage capacity.

A mainframe is a larger type of computer and is typically 10-100 times faster than the micro. These computers require a controlled environment both for temperature and humidity. Both the mini and mainframe computers will support more workstations than will a micro. They also cost a great deal more than the micro running into several hundred thousand dollars for the mainframes.

Workstations:

A **workstation** is a powerful Desktop computer designed for technical or scientific applications. Intended primarily to be used by one person at a time, they are commonly connected to a local area network and run multi-user operating systems. The term *workstation* has also been used to refer to a PC connected to a network.

Workstation



The characteristic features of workstation are:

Processing Power: The Processing Power of a workstation is several times more than that of an average power of a PC.

Storage capacity: Workstations have larger main memory (typically 0.5 GB to a few GB) as compared to PCs, which have few tens or hundreds of MB of main memory. The hard disk capacity of a workstation is also much more.

Display Facility: Most Workstations have a large screen (21 inch or more) monitor capable of displaying high-resolution graphics.

Processor design: PCs normally use CPUs based on CISC technology, whereas workstations use CPUs based on RISC technology. Popular RISC processors used in workstations are ALPHA.

Operating System: Unlike PCs which can run any of the five major OSs – MS-DOS, MS windows, Windows-NT.

Network interface card: Most Workstations have built-in hardware to connect to a LAN.

A computer that has a more power resources like Processing Power, Memory and additional capabilities to perform special task refers to as workstation. Workstations are often optimized for displaying and manipulating complex data such as 3D mechanical design, engineering simulation results such as for computational fluid dynamics, animation and rendering of images, and mathematical plots.

A workstations also fits on a desktop but has more powerful mathematical and graphics processing capability than a micro computer. Workstations are used for scientific, engineering, and design work that requires powerful graphics or computational capabilities. Workstations, 8 to 300 megabytes of RAM.

Review Questions:

1. Summarize the History of Computers.
2. Explain in detail about Generation of Computers.
3. Write short notes on the following:
RAM
ROM
EPROM
EEPROM
4. Sketch the block diagram of Computer.
5. Describe about I/O Devices.

Chapter II COMPUTER SOFTWARE

Aim:

Students are required to develop capacities in understanding and differentiating application and system software.

Objectives:

- Acquire knowledge on the past and new trends in computer software.
- Identify the various utility and software packages.
- Possession of basic skills to navigate windows operating system.

COMPUTER SOFTWARE: High level Languages, Low level language, assembly Language; translators-source program, object program, compiler, assembler, comparison of high and low level language and assembly languages, applications of BASIC, FORTRAN, COBOL, PASCAL, C, PROLOG, LOGO and PL/1.

(CAT II-PORION)

UTILITIES AND PACKAGES: Definition of Utilities and Packages; List of software available for data processing (dBase, FoxPro, RDBMS), word processing (WordStar, WORD, WordPerfect, regional language processor), Spreadsheet, DTP, CAD and bundled software's (MS-OFFICE'97).

OPERATING SYSTEM: What is an operating system? – Mode of operations-Batch, On-line, Time sharing, Real Time, PC operating systems-DOS, UNIX and WINDOWS NT. OS (Not for Examination)- Booting with DOS (from floppy disk / Hard disk), system prompt, changing default drive, files, and directories, Use of DOS commands: DATE, TIME, VER, CLS, DEL DIR (WITH /P/W OPTIONS and wild card characters only), CD, MD RD, PROMPT, TYPE, COPY, DEL, REN, FORMAT and PATH; redirecting output to printer; concepts of batch file (AUTOEXEC.BAT) and system file (CONFIG.SYS).

Computer Software: Languages

Low level and high level languages:

Machine and assembly languages are often referred to as low-level programming languages. These languages have the following limitations

1. They are machine dependent. A machine language program or an assembly language program cannot be executed on any computer other than the one for which it is written

2. They require the programmers to have a good knowledge of the internal structure of the computer being used.

3. It is difficult error prone and time consuming to write programs in machine language or assembly language, because they deal with machine-level coding, requiring one instruction to be written for each machine-level operation.

High- level programming languages were designed to overcome these limitations of low – level programming languages that is high level languages are characterized by the following features:

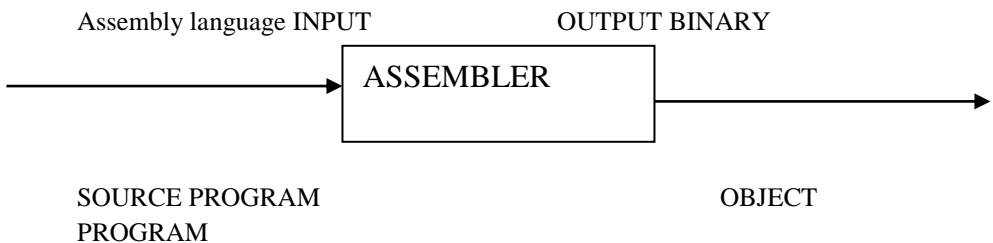
1. They are machine independent. That is, a program written in a high – level language can be easily ported and executed on any computer which has the translator software for the high – level language.

2. They do not require the programmer to know anything about the internal structure of the computer on which the high –level language programs will be executed.

3. They do not deal with the machine-level coding. Rather they deal with high-level coding enabling the programmers to write instructions using English words and familiar mathematical symbols and expressions.

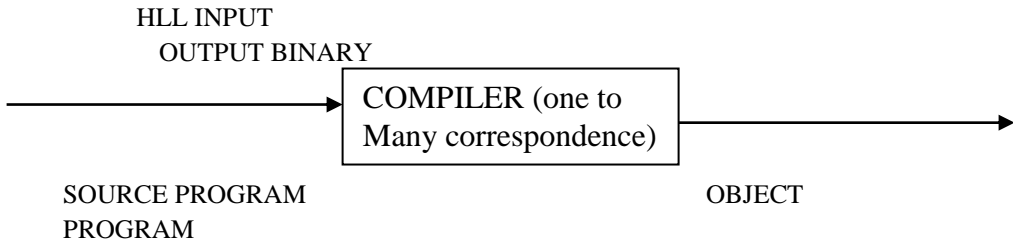
Assemblers:

Assembler: The process of translating an assembly language program into machine level language is called an assembler. The source program is not executed but only being converted into a form which can be executed by the computer processor.



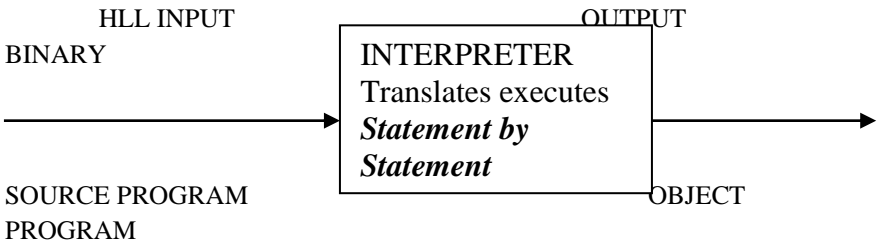
Compilers:

Compiler: It is system software; it is also called a translator. It converts high level language to its equivalent machine level language.



Interpreters:

Interpreter: It is another type of translator which is used for translating high level language program into its equivalent machine level language instructions and immediately executes the resulting machine language instructions.



Linkers:

Software often consists of several thousands, even several millions, of lines of program code. For software of this size, it is impractical to store all the lines of code in a single source program file due to the following reasons.

1. The large size of software would make it very difficult, if not impossible to work with.
2. It would make it difficult to deploy multiple programmers to work concurrently towards the development of the software for completing it within a specified time limit.

3. Any change in the source program, no matter how small, would require the entire source program to be recompiled. Recompilation of large source programs is often a time –consuming process.

System software is a type of program that acts like a conductor in an orchestra. It directs all the activities and sets all the rules for how the hardware and software work together. MS DOS and Microsoft Windows are examples of system software or operating system software.

Source Program

IS an original computer program written by a programmer that is converted in to the equivalent object program, written in machine language by the compiler or assembler?

A sequence of instructions that a computer can interpret and execute.

Object Program

A computer program translated from the equivalent source program in to machine language by the compiler or assembler.

A fully compiled or assembled program ready to be loaded.

COBOL Stands for Common Business Oriented Language. As its name implies it was designed for business data processing applications. Today it is the most widely used programming language for business-oriented applications. Business data processing applications deal with storing, retrieving, and processing corporate accounting information, and to automate such functions as inventory control, billing and payroll.

FORTRAN Stands for FORMulaTRANslation. It is one of the oldest high-level languages. It was designed to solve scientific and engineering problems, and is currently the most popular language amongst scientists and engineers.

BASIC Stands for Beginners All-purpose Symbolic Instruction Code. It is an interactive language which permits direct communication between the users and the computer, during the preparation and use of programs.

PASCAL

Pascal is widely recognized as a language, which encourages programmers to write well-structured, modular programs, and instills good programming practices in a programmer. Hence it is recognized as an education language, and is used to teach programming to beginners.

UTILITIES AND PACKAGES:

Types of software:

Computer software is broadly classified into two categories.

1. Application software
 - Software packages
 - Customized software

2. System software
 - Operating systems
 - Utilities
 - Language processors

Application software applications, Science and engineering applications, personal productivity program and so on. Personal productivity application programs are categorized based on the nature of their use in word processing, generating spreadsheet, presenting graphics and maintaining database. Application software's are available as packages as a collection of programs.

Software packages:

The general purpose application software is called a software package. Since the development of dedicated software for specific application is very expensive, software packages are developed to fulfill the needs of general computer users. The important categories of software packages are given below:

- Word processors
- Electronic spreadsheets
- Graphic, Multimedia, Presentation software
- Database software, etc.

Utilities: a set of programs, which help users in performing system maintenance and other tasks of routine nature.

Utility Program: a standard set of routines that assists in the operation of a computer system by performing some frequently required process such as sorting or merging.

Types of Utilities

- Disk formatting utility
- Data compression utility
- Data backup utility
- Antivirus utility

Disk formatting utility

Whenever a new disk is to be used with a system, it must be formatted properly according to the requirements of the associated operating system. The disk formatting utility allows user to perform this task with great ease.

Data compression utility

This utility uses a compression algorithm to transform files into a fraction of their normal size, so that they occupy less storage space on disk, or can be transferred across a network in less time. The utility has an associated decompression algorithm, which is used to bring back a compressed file into its normal form and size when accessed by a user. Often the amount of disk space saved by using this technique is significant – it can effectively double the capacity of a disk.

Data backup utility

Data stored in a computer system can be damaged or lost in several ways, such as a disk crash, a virus, a hardware malfunction, or simply an accidental erasure by its owner. Hence, it is always suggested to keep backup of important data. A backup utility is used to create copy of data on some storage media, such as floppy disk, CD-ROM, or magnetic tape, which can be stored off – line away from the computer system. When some data stored on – line is damaged or lost, it can be restored from the backup devices are kept on – line at a place away from the main computer system and the data backup is done automatically by the computer system on this device.

Antivirus utility

A computer virus is a piece of code attached to a legitimate program, which, when executed, infects other programs in the system by replicating and attaching itself to them. In addition to this replicating effect, a virus normally does some other damage to the system, such as corrupting / erasing files. Therefore, due to its spreading nature, a virus can cause severe damage to a system.

Utilities allow you to complete certain tasks on your computer. Examples of some of these tasks are file organizations. Specific purpose application software used to help a computer work better or to avoid problems.

Some utility programs are built into the operating system

- Scandisk in the Windows operating system
- Disk formatting software

Packages- A set of programs designed for a specific type of problem in statistics, production control etc. A written programs or procedures or rules and associated documentation pertaining to the operation of a computer system and that are stored in Read /Write memory.

Data Processing (dBase, Foxpro, RDBMS) / Database software:

Database refers to a collection of data stored in a desired order. for example, the names of students stored in the alphabetical order is a database. The data in the database can also be modified or indexed using key fields like Name, ID number, etc.

Database software sets up and maintains related data files. A data file contains records related to a person or organization. A database software has the capability to load, modify and update data. A database software helps to prepare and print details of employees, annual sales and so on. One of the major applications of a computer is Database Management system. The commonly available database management software includes Foxpro, SQL, Oracle, etc.



An Operating System (often referred to as OS) is an integrated set of programs that controls the resources (the CPU, memory, I/O devices, etc.) Of a computer system and provides its users with an interface.

dBASE: C. Wayne Ratiff is designed and programmed the first successful DBMS for personal computer. dBase format for storing data standard has become a de facto standard and is supported by all data base management and spreadsheet system.

Foxpro: Foxpro is a text based procedurally oriented programming language.

RDBMS: relational Data Base Management system. In RDBMS the data is structured in database tables, fields and records RDBMS. Each database row consists of one or more database table fields. RDBMS is the specific type, which follows the concepts of set theory or entity etc.

Word Processing (Word Star, WORD, Word Perfect, regional Language Processor) / Word Processor:

We use the word – processing software to prepare a document such as letter, memo or report. The document may be stored, edited, formatted or printed. A word processor is used to prepare a report, a personal or business letter. Word processors allow the user to format of the document. The commonly used word processor software includes MS Word, Wordpad, Wordstar, etc.



Word Star: Word Star is the word processor application. Word star created for windows in 1991.

WORD: a word making a paragraph and entire pages.

Word Perfect: Word Perfect is a word perfect application, now owned by corel. Word perfect 1.0 represented a standard word processing. The common file name extension of word perfect document file is .wpd.

Regional language Processor: using Regional Language Processor options in control panel, you can install multiple languages on your computer, such as Arabic, Japanese, Korean, Western, European languages etc.

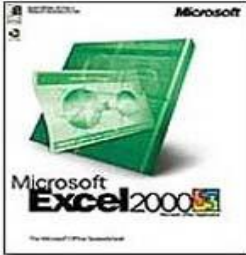
Features in MS Word:

- Entering Text
- Editing Text
- Basic Formatting
- Page layout and paragraph formatting
- Numbering and bullets
- Spelling
- Mail merge
- Previewing your document

Electronic spreadsheet:

A spreadsheet is a simple worksheet comprising rows and columns of data. The data kept in a spreadsheet is similar to the data written in a ledger book of a band or the mark sheet of a student.

Electronic spreadsheet software is used to prepare documents containing information or data in the form of numbers or characters that are arranged in rows and columns for further processing and analysis, preparing reports and generating charts. The spreadsheet software is also capable of performing arithmetic operations and using functions.



Data stored in the spreadsheet can be used to prepare the budget or monthly income statement; a teacher can store marks to maintain a grade book, etc. since the spreadsheet software can perform arithmetic operations using an equation, the user can easily analyze the data. The commonly available electronic spreadsheet software includes Excel, Lotus 1-2-3, etc. Microsoft Excel is popular electronic spreadsheet software. The Excel worksheet has many parts, some of which are listed below:

- Rows and columns
- Cells
- Cell Contents
- Spreadsheet commands
- Graphs in spreadsheet.

DTP: Desktop Publishing: DTP is the use of the computer and software to create visual displays of ideas and information. Desktop publishing documents may be for desktop or commercial printing or electronic distribution including PDF, slide Shows, e mail newsletter and the Web.



CAD / Graphic software:

Graphic software helps to prepare automated drawings using the computer and peripheral devices. The drawings may be stored, edited or printed.

Graphic packages are widely used in video games, multimedia applications, computer aided design (CAD) is also known as Computer Aided Drafting, Computer Assisted Design, Computer aided engineering (CAE) and so on. These packages are used to generate drawings, pictures and charts. The geometrical model can be developed using CAD. Using CAE, results of analysis can be graphically obtained, e.g. bar charts, pie charts, and so on.

Graphic software can draw a graph, bar chart, etc. some graphic software enables the user to draw illustrations. Printers or plotters are used to print the computer – generated drawings on film or paper.

Bundled Software's (MS – OFFICE '97):

Microsoft office 97 was a major milestone release of Microsoft office, which included hundreds of new features and improvements, introduced "Command Bars", a paradigm in which menus and tool bars were made more similar in capability and visual design featured natural language systems and sophisticated grammar checking.

MS OFFICE 97 – Excel, outlook, word 97, access and power point.

OPERATING SYSTEM:

Operating System: an integrated set of programs, which controls the resources of a computer system, and provides its users with an interface or virtual machine, which is more convenient to use than the bare machine. Operating system is the Master Program that instructs computer how to work with its various components and resources.

Definition: an operating system is a collection of programs which act as an interface between a user of a computer and the computer hardware. It is also called as resource allocator.

Purpose of an operating system:

To provide an environment in which a user may execute programs. It also manages the various resources of the computer such as processor, storage, input and output devices, etc. it also acts as a user interface.

Mode of operation:

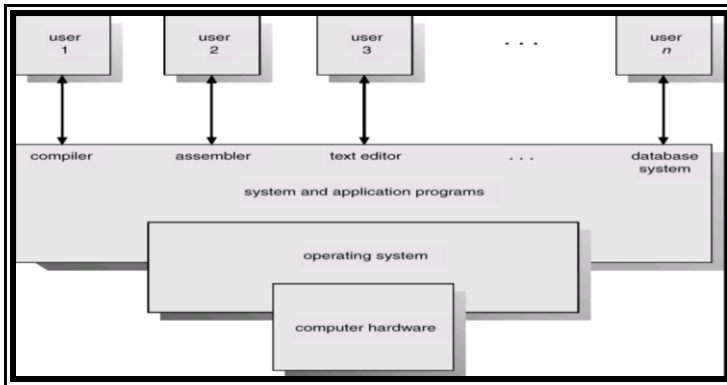
Operating system controls and co-ordinates the function of the computer system. Disk operating system is kind of operating system which operates through either a floppy disk or hard disk. Provide the instructions to prepare user interface.

Computer Software:

1-System Programs (OS)

2-Application Programs

- Operating system controls all computer resources and provides the base upon which the application programs can be written
- Operating system is the software layer that is on top of the hardware to manage all parts of the system, and present the user with interface or virtual machine that is easier to understand and program.



Need of an operating system:

- The main need of an operating system is making a computer system convenient to use.
- It acts as an interface between the hardware and its users, providing a high level of interface to low level hardware, and making it easier for the programmers and other users to access and use those resources.

Types of operating system:

- Batch Operating System (Single user operating system)
- Online operating system.
- Time sharing systems.
- Real time system

Batch operating system (Mid 1950's – Mid 1960's):

- A batch system is one in **which jobs are bundled together with the instructions** necessary to allow them to be processed without involvement.
- **The monitor is system software that is responsible for interpreting and carrying out the instructions in the batch jobs.**

When the monitor starts a job, the entire computer is dedicated to the job, which then controls the computer until it finishes.

- **All the resources are allocated to a single user operating system is called a single user operating system. Examples of single user Operating System:- MS-DOS (Microsoft Disk Operating System) Windows 98, Windows 2000, Windows XP.**

Batch Processing: the running of several computer programs one after another without the need of a human operator to run each program individually.

Advantages of batch systems:

- Move much of the work of the operator to the computer
- Increase performance

Disadvantages:

- more difficult to debug(Correct) program
- due to lack of protection scheme, one batch job can affect pending jobs (read too many cards, etc)
- a job could corrupt the monitor, thus affecting pending jobs
- a job could enter an infinite (endless)loop

Online Operating System:

- **An operating system that runs in a server on the Web.**
- The environment created in a user's machine from an **online application stored on the Web and run through a Web browser**
- This processing is also used for efficient start database updates and automated transaction Processing.

Real Time Operating System:

A real time operating system is **multitasking**. **Multitasking:** More than one job at the state of execution, **jobs submitted by the same user**.

Real Time Systems:

- A primary objective of real time systems is to **provide quick event-response times**.

- A real time system is often **used as a control device in a dedicated application.**

Such applications include:

- Industrial control.
- Telephone switching system.
- Flight control.
- Military applications.
- A real time operating system has well defined fixed time constraints.

Time sharing system:

- **A Time-shared operating system allows many user to simultaneously share the computer.**
- Since each action or command in a time shared system tends to be short only a little CPU time is needed for each user.
- It uses multiprogramming with special CPU scheduling algorithms to achieve this.
- **(Multiprocessing: If a system has more than one processor (CPU) it is said to be a Multiprocessing system.)**
- The multiprogramming feature allows multiple programs to simultaneously reside in the main memory. Each user is given a time slot of (10 milliseconds) to (100 milliseconds).

Advantages:

- Reduces CPU idle time.
- Provides advantages for quick response time.
- Offers quick/good computing facilities to small users.
- Time sharing is the sharing of a computing resource among users by means of **Multiprogramming and Multiprocessing.**

PC operating systems – DOS, UNIX and WINDOWS NT:

MS-DOS –Microsoft Disk Operating System :

- MS- DOS Stands for Microsoft disk operating system. It is a single user operating system for IBM and IBM-compatible personal

computers. It was introduced in 1981 jointly by Microsoft and IBM and was the most popular operating system for personal computers in the 1980s. Its popularity started reducing in the 1990s with the launch of Microsoft windows operating system.

Command	Description
Cd	Changes directories
Cls	Clears the screen.
Cmd	Opens the command interpreter.
Lock	Lock the hard disk drive
Help	Display a list of commands and brief explanation.
Exit	Exit from the command interpreter.
Erase	Erase files from computer.
Edit	View and edit files.
Del	Deletes one or more files.
Date	View or change the systems date.
Copy	Copy one or more files to an alternate location.

UNIX:

- Unix was developed in the early 1970s at Bell Laboratories by Ken Thompson and Dennis Ritchie for a small PDP-11 Computer. it was the first operating system to be written in a high-level language, C. the normal

Basic UNIX commands

The below commands are some basic UNIX commands

Ls--- Lists your files

Date --- Shows the current date and time.

Cal --- Shows a calendar of the current month

Mkdir*dirname*--- Make a new directory

Cdd*dirname*--- Change directory

mv*filename1 filename2*--- Moves a file

cp*filename1 filename2*--- Copies a file

rm*filename*--- Removes a file

diff*filename1 filename2*---Compares files

ls -l --- Lists your files in 'long format'

ls -a --- Lists all files

wcfilename--- Tells you how many lines, words, and characters there are in a file

Windows NT:

Microsoft Windows NT is a multi – user, time sharing operating system developed by Microsoft. It was designed to have Unix- like features so that it can be used for powerful workstations, networks and database servers. its main features are as follows:

- Unlike UNIX, its native interface is a GUI. The look and feel of Microsoft Windows NT’s GUI is similar to that of Microsoft window’s GUI.
- It supports multiprogramming and is also designed to take advantage of multiprocessing on systems having multiple processors.
- It has built – in networking and communications features, so that any computer with Microsoft Windows NT can be made to work as a network client or server.
- It provides strict system security
- It has a rich set of tools for software development and system administration.
- It can run Microsoft windows applications and many UNIX applications directly.

BOOTING WITH DOS (FLOPPY DISK/ HARD DISK):

The term booting means start or start-up of a computer. When you boot your computer, you are turning it ON and loading the operating system into your computer’s memory. This includes reading the boot record into memory, which passes control to DOS checks the presence of IO.SYS and MS DOS.SYS and COMMAND.COM. If these files are found the DOS prompt is displayed on your screen. This prompt, normally an A :>, B:>, or C:>, depending on which disk drive is active, invites you to type a command.

Booting checks the following DOS Programs. Because these program form the basis of operation of a computer.

The IO.SYS Program: IO.SYS contains at least five device drivers, which are used to control the operation of keyboard, screen, printer, disk controller, and system clock. Other device drivers are used to control large external hard disks, serial communications, joy sticks and mouse devices.

The MSDOS.SYS Program: It interacts directly with application programs and the IO.SYS program described above. It intercepts program calls for printing, data storage or retrieval, information display, and so on.

The COMMAND.COM Program: the COMMAND.COM program is called the command processor. It intercepts or reads commands from the keyboard and executable command files.

If anything is wrong with these files, the computer can't be started.

There are two types of booting: (1) cold boot and (2) warm boot.

(1) COLD BOOT:the cold boot is that situation when all the computer peripherals are off and start it by switching on the power.

(2) WARM BOOT: in this case the computer and its related peripherals are already switched on. We restart the system by pressing the RESET button on the front panel of CPU or by pressing the combination of CTRL + ALT+ DEL keys from the keyboard. In warm boot we don't have to switch off the system. If the computer "hangs" (or crashes) due to some reason it may be necessary to reset with CTRL + ALT+ DEL or by pressing reset button to get it going again.

Changing default drive, files and directories:

DOS FILES:

In DOS, programs and data are stored in files. A File is collection of information and it the primary storage unit of data storage in DOS-based system. A DOS File can contain either program instructions or data. Each DOS file has a unique file name. file name is a unique name used to identify a file. A file has two parts: 1. First name 2. Extension name.

Example:

REPORT.DOC First name Extension name

The first name can be up to 8 characters long.

The extension name is optional and up to 3 characters long. This is used to indicate the type of every file.

The first and the extension names are separated by a dot (.). There is no space before or after the dot (.)

The autoexec.bat file:

When we boot the computer we need to set the prompt, path and other settings. Instead of typing this again and again, we can store these settings in a batch file called AUTOEXEC.BAT. DOS will automatically execute these commands during boot-up.

The AUTOEXEC.BAT file is located in the root directory and this file is read automatically when we boot the computer. This file executes a sequence of commands, like setting the path and prompt etc.

For Ex, if we want to include the commands for changing the prompt on our screen, setting the path for DOS and wordstar and then clearing the screen, type the following commands

C:\>COPY CON AUTOEXEC.BAT

After pressing the <ENTER> key, the cursor will appear on the next line, now we can type

```
PROMPT $P$G
PATH=C:\DOS; C:WS
CLS
```

After we have finished entering the commands, press F6. Now each time we switch on the computer the prompt(C:) will be changed so that it displays the current directory. The PATH command tells DOS to look in the DOS and WS directories for program files. At the end, the screen will be cleared by the CLS command.

Uses of Batch files:

DOS will automatically either these following commands during boot-up.

Ex:

Create a batch file for the following:

Copy all the files having an extension.TXT from the root directory to D:\VRB

Copy all the files having an extension.EXE from the root directory to D:\VRB

Copy all the files having an extension.BMP from the root directory to D:\VRB

Delete all the files having an extension.BAK.

```
C:\>COPY CON EXBAT.BAT
COPY*.TXT D:\VRB
COPY*.EXE D:\VRB
COPY*.BMP D:\VRB
DEL *.BAK
```

Press `Z or F6 to save batch files and return to prompt. Now simply type the name of the batch file in the command prompt for running.

```
C:\>ExBAT
```

And press <Enter> key. It executes the above specified DOS commands sequentially.

Difference between MS DOS and WINDOWS:

S.N	MS-DOS	WINDOWS
	It is a Character User Interface	It is a Graphical User Interface
	To execute commands, the user has to type from the keyboard. The command must be in exact syntax	To execute commands, the user has to clicks from pull down menu or click the appropriate icons using mouse.
	It is single user operating system	It is multi user operating system.
	Not possible to work on many operation, simultaneously.	It is possible to work on many applications simultaneously(Multitasking Operating System)
	In the most DOS based applications, mouse operations are not available.	Using mouse, we can run any application

MICROSOFT WINDOWS:

WINDOW: A rectangle portion of a computer screen, which is dedicated to specific activity or application. The display screen can be separated into several windows to simultaneously show the status of the different applications at any instance of time.

Windows : an operating system developed by Microsoft for use on personal computers. Its various release versions are windows 3.0, 95, 98, 2000, 2003 and windows Xp.

It provides Multitasking capabilities to the personal computer. It runs each program or document or application in its own separate window. Windows have the facility of accessories such as word pad, note pad, calculator and paint. Users can paste portion one document into another by utilizing more advanced document linking features called DDE(Dynamic data Exchange), OLE(Object linking and Embedding).

GUI: (Graphical user Interface)an interface for computer users, which provides icons(pictures) and menus(list of choices), which users can select with a mouse for telling the computer what they want it to do for them.

FUNCTIONS OF WINDOWS:

TYPES OF WINDOWS

Working with applications, two types of windows appear on the desktop: Application windows and windows inside the application window. An application window contains the running application. The Application window has a title bar and menu bar. Another window called document window, may appear inside an Application window. This type of window may contain documents, Groups (in Program Manager), or directories.

BASIC TECHNIQUES FOR WORKING IN WINDOWS

Using a mouse is usually easier and faster than using the keyboard but one need to know both mouse and keyboard techniques to work in Windows. Almost all the functions in Windows can be performed using either the keyboard or a mouse.

Basic Mouse Techniques

Generally the mouse has two buttons where one is the primary mouse button and the other is secondary. In Windows, the left mouse button is set as the primary button by default. The primary button is used to perform the majority of the tasks whereas the secondary button is manipulated by few of the windows-based applications only.

Control-Menu box:

It is in the upper-left corner of each window. Clicking on the Control-Menu box opens the Control menu. The Control menu can be used To move, size, and close a window while working with the Keyboard.

TYPES OF ICONS

The small graphical pictures that represent various types of applications and files are called Icons.

There are three types of icons:

Type	Description
Program-item icon	The Program-item icon can be moved from one group window to another.
Application icon	Represents an application that is running but whose window is reduced to an icon.
Group Icon	Represents the application that can be started from Windows.

These icons can be found under the groups in the Program Manager Menu Bar

It lists the available menus. A menu contains a list of actions or Commands

What Is Windows XP:

Windows XP is the latest and greatest version of the Windows operating system.

The *XP* stands for experience (as in “The Jimi Hendrix . . .” for those of you who were around back then). This version of Windows is certainly going to offer a new experience for those of you coming from earlier versions of Windows. It’s a type of program known as an *operating system*, often abbreviated as OS. You might be aware that Windows XP comes in several different flavors, two of which are *client-side* products: The Windows XP Home Edition, and the Windows XP Professional Edition. The other versions are *server-side* products, named Windows XP Server and Windows XP Advanced Server.

Microsoft Windows NT:

Microsoft Windows NT is a multi-user, timesharing operating system developed by Microsoft, Main features are as follows:

It supports multiprogramming and is also designed to take advantage of multiprocessing on systems having multiple processors.

It has built-in networking and communications features, so that any computer with Microsoft Windows NT can be made to work as a network client or server.

It provides strict system security. It has a rich set of tools for software development and system administration.

Screen Elements

Title bar – This is the title of the window, like your name at the top of a piece of paper. The Title bar is also the handle for the window. If you click and hold the mouse button down on the title bar, you can move it around the screen.

Minimize – To have a window take the minimum amount of desktop space possible, click the

Minimize button. This drops the window into the Task bar like a piece of paper going into aDrawer. The Task bar will show the task whether or not the window is minimized.

Maximize – To have a window take the maximum desktop space, click the maximize button.This stretches the window out like an architect’s floor plan being rolled out over the desk.

Restore – A maximized window will cover over all the other windows and icons on the desktop. The Restore button places the window back so that more than one window can display at a time.

Restore down: which returns the window to its smaller size.

Close (X) – When done with a window, you can have it taken completely off the desktop byclosing it. Use the X button to do this. Close the window

Menu bar: The Menu bar displays a list of commands that can be used to perform various tasks. (This will be below the Title bar). Menu items are commands within the menu bar that allow choosing of functions & tasks.

Tool bar: contains a set of buttons for frequently used commands.

Address Bar: a convenient way of accessing the various parts of the system.

Scroll Bars: allow you to view information which is outside the current window area. Both vertical and horizontal bar may be provided. On the bottom and right edges of a window we find scroll bars. They are used to pan across the information in the window, when we have information which won’t fit into the window.

Status Bar :The Status bar appears at the very bottom of the window and provides such information as the cursor position, current page number, the number of words in the document etc.

List of programs available in MAIN, Accessories, Games and applications and group

Windows:

Accessories:

Calculator, command prompt, connect to a network projector, connect to a projector, getting started, math input panel, notepad, paint, remote desktop connection, run, snipping tool, sound recorder, sticky notes,

sync center, windows explorer, windows mobility center, word pad, ease of access, system tools.

Games:

Games explorer.

Maintenance:

Backup and restore, create a system repair disc, help and support and windows remote assistance.

Operating Windows:

Mouse: The mouse is a small device used to point to a particular place on the screen and select in order to perform one or more actions. It can be used to select menu commands, size windows, start programs etc. The most conventional kind of mouse has two buttons on top: the left one being used most frequently.

INTERNET&WEB BROWSING

INTERNET

The Internet was created by the Advanced Research Projects Agency (ARPA) and the U.S. Department of Defense for scientific and military communications. The Internet is a network of interconnected networks. Even if part of its infrastructure was destroyed, data could flow through the remaining networks. The Internet uses high-speed data lines, called backbones, to carry data. Smaller networks connect to the backbone, enabling any user on any network to exchange data with any other user.

Major Features of the Internet:

- The World Wide Web
- E-Mail
- News
- Telnet
- File Transfer Protocol (FTP)
- Internet Relay Chat (IRC)

The protocol used by the Internet is the Transmission Control Protocol/Internet Protocol, or TCP/IP. No matter what type of computer system you connect to the Internet, if it uses TCP/IP, it can exchange data with any other type of computer.

Major Features of the Internet:

The World Wide Web is a part of the Internet, which supports hypertext documents, allowing users to view and navigate different

types of data. A Web page is a document encoded with hypertext markup language (HTML) tags. HTML allows designers to link content together via hyperlinks. Every Web page has an address, a Uniform Resource Locator (URL).

Web Browsers:

One of the first problems → web Browsers were originally intended only to display static documents, returned by HTTP calls

Difficult to build sophisticated application specific clients for web browsers.

INTERNET EXPLORER

Windows Internet Explorer (formerly **Microsoft Internet Explorer**, commonly abbreviated **IE** or **MSIE**) is a series of graphical web browsers developed by Microsoft and included as part of the Microsoft Windows line of operating systems starting in 1995.

IE has been the most widely used web browser since 1999, attaining a peak of about 95% usage share during 2002 and 2003 with IE5 and IE6.

Since its first release, Microsoft has added features and technologies such as basic table display (in version 1.5); XMLHttpRequest (in version 5), which aids creation of dynamic web pages; and Internationalized Domain Names (in version 7), which allow Web sites to have native-language addresses with non-Latin characters. The browser has also received scrutiny throughout its development for use of third-party technology (such as the source code of Spyglass Mosaic, used without royalty in early versions) and security and privacy vulnerabilities, and both the United States and the European Union have alleged that integration of IE with Windows has been to the detriment of other browsers.

The latest stable release is Internet Explorer 9, which is available as a free update for Windows 7, Windows Vista, Windows Server 2008 and Windows Server 2008 R2

i. NETSCAPE NAVIGATOR

Netscape Navigator is a proprietary web browser that was popular in the 1990s. It was the flagship product of the Netscape Communications Corporation and the dominant web browser in terms of usage share, although by 2002 its usage had almost disappeared. This was primarily

due to the increased usage of Microsoft's Internet Explorer web browser software, and partly because the Netscape Corporation (later purchased by AOL) did not sustain Netscape Navigator's technical innovation after the late 1990s.^[1]

The Netscape Navigator web browser was succeeded by Netscape Communicator.

Control Panel: Add new hardware: add a new software, printer installation, date/time , mouse, and regional settings:

Installing Hardware:

1. Click the start button, point to settings and then click control panel.
2. Double click add new hardware.
3. Follow the instruction on your screen.

Installing Software:

1. Click the start button, point to setting and then click control panel.
2. You can add or remove components of windows by clicking the windows set up tab in the Add/Remove Programs Window.
3. Double click Add/Remove Programs.

Printer Installation:

Before you begin, make sure your printer is correctly connected to your computer and that you know the make and model of the printer. If you're planning to use a shared printer over the network, you may also need to know the path to it. Or you can browse for the printer in Network Neighborhood and then double-click its icon to set it up.

To set up a printer for use with windows, do the following:

1. Click the start button, point to settings and then click printers. If any printers are already set up, their icons appear in the Printers folder. These printers are ready for you to use.
2. Double click Add Printer.

3. Follow the instructions on your screen. You can use these same steps to set up a shared network printer. Just click Network Printer when the wizard prompts you.
4. When you finish, the icon for your printer appears in the printers folder. Your printer is ready for you to use. If you set up both local and shared network printers, notice that the icons are different.
You can double click the printer icon in the printers folder to see the documents that are printing or waiting to print and to manage printing. For example, you can pause or cancel the printing of one or more documents.

Review Questions:

1. Differentiate High level, low level and Assembly level languages.
2. Write down the applications of BASIC, FORTRAN, COBOL, PASCAL , C, PROLOG, LOGO and PL/1 languages.
3. Summarize Utilities and packages of computer.
4. Paraphrase about the functioning of an operating system.
5. Explain in Detail about Microsoft windows.
6. Differentiate the following browsers:
 - Mozilafirefox
 - Internet Explorer
 - Netscape Navigator

Chapter III

DATA PRESENTATION

Aim:

Students are required to have competence in the prerequisites of data presentation inside the computer system and program development.

Objectives:

- Ability to formulate and solve the different number representations.
- The skill to design and develop programs using different computing methodologies.

Data Presentation inside the Computer:

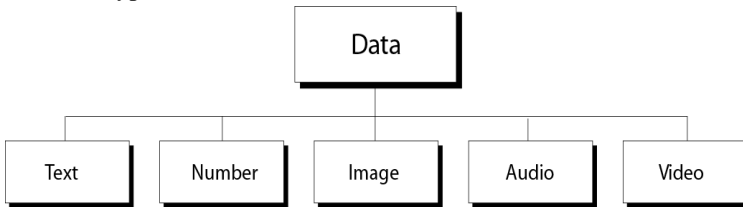
Binary, Octal and hexadecimal number systems with conversions; Binary Addition and Subtraction-Simple Problems; ASCII, BCD and EBCDIC codes.

Program Development

Problem Definition, Algorithm Flowchart, Coding, Compiling and Debugging and Running.

DATA PRESENTATION

Different types of data



■ Data comes in different forms such as numbers, text, images, audio, video,

■ How does the computer handle all the different data types?

■ The most efficient solution is to use a uniform representation of data. All data from outside a computer are transformed into this uniform representation.

■ The universal format is called a *bit pattern (a sequence of bits)*.

■ How many bits are needed in a bit pattern to represent a symbol in a language?

■ A bit pattern of size 1 can represent two different pieces of information. For example, to represent pass / fail we could use 0 - Fail

1 - Pass

■ A bit pattern of size 2 can represent four different pieces of information. For example, to represent the four undergraduate classifications, we could use

00 - freshman

01 - Sophomore

10 - Junior

11 - Senior

Binary integer arithmetic

Integers are stored in a computer using a fixed number of binary digits to encode each value that is 0 and 1s.

Bases other than 2 and 10

■ Any positive integer can be used as a base.

■ For base 5

■ The digits are {0, 1, 2, 3, 4}

■ In a number, such as 1203, each digit is implicitly multiplied by a positional power of 3.

■ 1203 base 5 is

NUMBER SYSTEM

A computer understands information composed of only 0's and 1's.

Definition: Number system is the way to represent a number in different forms.

Basic Number Systems and Conversions

When we write decimal (base 10) numbers, we use a positional notation system. Each digit is multiplied by an appropriate power of 10 depending on its position in the number:

For example:

$$\begin{aligned} 843 &= 8 \times 10^2 + 4 \times 10^1 + 3 \times 10^0 \\ &= 8 \times 100 + 4 \times 10 + 3 \times 1 \\ &= 800 + 40 + 3 \end{aligned}$$

For whole numbers, the rightmost digit position is the one's position ($10^0 = 1$). The numeral in that position indicates how many ones are present in the number. The next position to the left is ten's, then hundred's, thousand's, and so on. Each digit position has a weight that is ten times the weight of the position to its right. In the decimal

number system, there are ten possible values that can appear in each digit position, and so there are ten numerals required to represent the quantity in each digit position.

The decimal numerals are the familiar zero through nine (0, 1, 2, 3, 4, 5, 6, 7, 8, 9).

In a positional notation system, the number base is called the radix. Thus, the base ten system that we normally use has a radix of 10. The term radix and base can be used interchangeably. When writing numbers in a radix other than ten, or where the radix isn't clear from the context, it is customary to specify the radix using a subscript. Thus, in a case where the radix isn't understood, decimal numbers would be written like this: 12710 1110 567310

Generally, the radix will be understood from the context and the radix specification is left off. The binary number system is also a positional notation numbering system, but in this case, the base is not ten, but is instead two. Each digit position in a binary number represents a power of two. So, when we write a binary number, each binary digit is multiplied by an appropriate power of 2 based on the position in the number:

For example:

$$\begin{aligned} 101101 &= 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 1 \times 32 + 0 \times 16 + 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 \\ &= 32 + 8 + 4 + 1 \end{aligned}$$

In the binary number system, there are only two possible values that can appear in each digit position rather than the ten that can appear in a decimal number. Only the numerals 0 and 1 are used in binary numbers. The term 'bit' is a contraction of the words 'binary' and 'digit', and when talking about binary numbers the terms bit and digit can be used interchangeably. When talking about binary numbers, it is often necessary to talk of the number of bits used to store or represent the number. This merely describes the number of binary digits that would be required to write the number. The number in the above example is a 6 bit number.

Types of Number system:

1. **Binary Number System:** It is the number system with base value 2 means it has only two digits to represent the data. The digits are (0, 1). E.g. 00,01,10,11,100....
2. **Decimal Number System:** It is the number system with base value 10 means it has 10-digits to represent the data. The digits are(0-9). Eg. 0,1,2,3,4,5,6
3. **Octal Number System:** It is the number system with base value 8 means it has 8 digits to represent the data. The digits are (0-7).
4. **Hexadecimal Number System :** It is the number system with base value 16 means it has 16 digits to represent the data. The digits are (0-15). Eg. 0,1,2,3.....,9,A,B,C,D,E,F

Binary

- Base or radix 2 number system
- **Binary digit** is called a bit.
- Numbers are 0 and 1 only.
- Numbers are expressed as powers of 2.

Conversion of binary to decimal (base 2 to base 10)

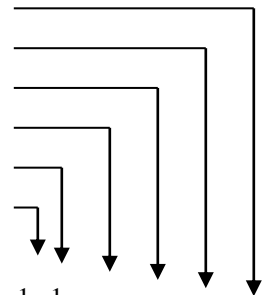
Example: convert $(110011)_2$ to decimal

$$\begin{aligned} &= (1 \times 2^5) + (1 \times 2^4) + (0 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + \\ &(1 \times 2^0) \\ &= 32 + 16 + 0 + 0 + 2 + 1 \\ &= (51)_{10} \end{aligned}$$

Conversion of decimal to binary (base 10 to base 2)

Example: convert $(51)_{10}$ to binary

$$\begin{aligned} 51 \div 2 &= 25 \text{ remainder is } 1 \\ 25 \div 2 &= 12 \text{ remainder is } 1 \\ 12 \div 2 &= 6 \text{ remainder is } 0 \\ 6 \div 2 &= 3 \text{ remainder is } 0 \\ 3 \div 2 &= 1 \text{ remainder is } 1 \\ 1 \div 2 &= 0 \text{ remainder is } 1 \end{aligned}$$



Answer = 1 1 0 0 1 1

Note: the answer is read from bottom (MSB) to top (LSB) as 110011_2

Octal

- Base or radix 8 number system
- 1 octal digit is equivalent to 3 bits.
- Numbers are 0-7.

Conversion of octal to decimal (base 8 to base 10)

Example: convert $(632)_8$ to decimal

$$\begin{aligned} &= (6 \times 8^2) + (3 \times 8^1) + (2 \times 8^0) \\ &= (6 \times 64) + (3 \times 8) + (2 \times 1) \\ &= 384 + 24 + 2 \\ &= (410)_{10} \end{aligned}$$

Conversion of decimal to octal (base 10 to base 8)

Example: convert $(177)_{10}$ to octal

$$177 \div 8 = 22 \text{ remainder is } 1$$

$$22 \div 8 = 2 \text{ remainder is } 6$$

$$2 \div 8 = 0 \text{ remainder is } 2$$

Note: the answer is read from bottom to top as $(261)_8$, the same as with the binary case.

Hexadecimal

- Base or radix 16 number system
- 1 hex digit is equivalent to 4 bits.
- Numbers are 0-9, A, B, C, D, E, and F.
- $(A)_{16} = (10)_{10}$,
- $(B)_{16} = (11)_{10}$,
- $(C)_{16} = (12)_{10}$,
- $(D)_{16} = (13)_{10}$,
- $(E)_{16} = (14)_{10}$,
- $(F)_{16} = (15)_{10}$
- Numbers are expressed as powers of 16.

Conversion of hexadecimal to decimal (base 16 to base 10)

Example: convert $(F4C)_{16}$ to decimal

$$\begin{aligned} &= (F \times 16^2) + (4 \times 16^1) + (C \times 16^0) \\ &= (15 \times 256) + (4 \times 16) + (12 \times 1) \end{aligned}$$

$$= 3840 + 64 + 12$$

$$= (3916)_{10}$$

Conversion of decimal to hex (base 10 to base 16)

Example: convert $(77)_{10}$ to hex

$$77 \div 16 = 4 \text{ remainder is D}$$

$$4 \div 16 = 0 \text{ remainder is 4}$$

Note: the answer is read from bottom to top as $(4D)_{16}$, the same as with the binary case.

Decimal	Binary	Octal	Hexadecimal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Table of Binary, Decimal and Hexadecimal Numbers

Conversion of Octal and Hex to Binary

Conversion of octal and hex numbers to binary is based upon the bit patterns shown in the table above and is straight forward. For octal numbers, only three bits are required.

Thus $6_8 = 110_2$, and

$$345_8 = 111\ 001\ 012.$$

For hex numbers, four bits are required.

Thus $E_{16} = 1110_2$, and

$$47D_{16} = 100\ 0111\ 1101_2.$$

Conversion of Binary to Octal and Hex

Conversion of binary numbers to octal and hex simply requires grouping bits in the binary numbers into groups of three bits for conversion to octal and into groups of four bits for conversion to hex. Groups are formed beginning with the LSB and progressing to the MSB.

Thus, $11\ 100\ 111_2 = 347_8$ and

$$11100010101010010001_2 = 7025221_8.$$

Similarly,

$$1110\ 0111_2 = E7_{16}$$
 and

$$1\ 1000\ 1010\ 1000\ 0111_2 = 18A87_{16}.$$

Binary Arithmetic

Binary Addition

It is a key for binary subtraction, multiplication, division. There four rules of the binary addition.

In fourth case, a binary addition is creating a sum of $(1+1=10)$ i.e. 0 is write in the given column and a carry of 1 over to the next column.

Case	A	+	B	Sum	Carry
1	0	+	0	0	0
2	0	+	1	1	0
3	1	+	0	1	0
4	1	+	1	0	1

Binary Addition Table

The entry for 1+1 is 10 which indicates a carry of 1

<i>Examples</i>	<i>Step 1</i>	<i>Step 2</i>	<i>Step 3</i>
<i>Step 4</i>	<i>Step 5</i>		
Addend 1011 1011	1011	1011	1011
Augend + 100 + 100	+ 100	+ 100	+ 100
Sum		1	11
1111			111

<i>Examples</i>	<i>Step 1</i>	<i>Step 2</i>	<i>Step 3</i>
<i>Step 4</i>	<i>Step 5</i>	<i>Step 6</i>	
carry		1	1
1	1		
Addend 1101 1101	1101	1101	1101
Augend + 1001 + 1001	+ 1001	+ 1001	+ 1001
Sum	+ 1001	0	10
0110	10110		110

Binary Subtraction

Subtraction and Borrow, these two words will be used very frequently for the binary subtraction. There four rules of the binary subtraction. There four rules of the binary Subtraction.

Case	A	-	B	Subtract	Borrow
1	0	-	0	0	0
2	1	-	0	1	0
3	1	-	1	0	0
4	0	-	1	0	1

Binary Subtraction Table

<i>Examples</i>	<i>Step 1</i>	<i>Step 2</i>	<i>Step 3</i>
<i>Step 4</i>	<i>Step 5</i>	<i>Step 6</i>	
borrow	01	01	01
0	0		
Minuend	10100	10 1 00	10 1 00
Subtrahend	10 1 00	1 0100	1 0100
	- 1001	- 1001	- 1001
	- 1001	- 1001	- 1001
Difference	<u> </u>	<u> </u>	<u> </u>
011	1011	01011	

Note: This problem in decimal is $20 - 9 = 11$ which is the answer we get in binary.

Example - Addition

$$\begin{array}{r}
 0011010 + 0011100 = 00100110 \\
 + 0011100 \\
 \hline
 0100110 = 38_{10}
 \end{array}$$

Example - Subtraction

$$0011010 - 001100 = 00001110$$

$$\begin{array}{r} 11 \text{ borrow} \\ 00\cancel{1}1010 = 26_{10} \\ -0001100 = 12_{10} \\ \hline 0001110 = 14_{10} \end{array}$$

BCD-(Binary coded decimal Approach)

With BCD, it is possible to convert each decimal number into its binary equivalent rather than convert the entire decimal value into a pure binary form. BCD equivalent of each possible decimal symbol is shown below.

Since 8 and 9 require 4 bits, all decimal digits are represented in BCD by 4 bits.

When 4 bits are used, there are only 16 possible configurations (2^4).

As shown below, the first 10 of these combinations are used to represent decimal digits.

The other six arrangements (1010, 1011, 1100, 1101, 1110, and 1111) have decimal values from 10 to 15.

0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
DEC	8	4	2	1

This arrangement is not used in BCD coding. That is, 1111 does not represent 15_{10} . Instead, the proper BCD code for 15_{10} is 0001/0101.

ASCII and EBCDIC

ASCII (American standard code for information Interchange)

Computers use binary digits that can only have two values called 0 and 1. A bit is a single binary digit. It represents the smallest unit of data (just like the good old American penny). However, computers usually do not operate on single bits; rather they store and manipulate a fixed number of bits. Most often, the smallest unit or number of bits a computer works with is eight bits. These eight bits make up a byte. ASCII codes use eight bits, and those eight bits represent a single character, such as the letter A or the number 7. Thus, the computer can store and manipulate an individual byte (a single character) or a group of bytes (several characters, a word) at a time. These individual bytes, or groups of bytes, form the basic unit of memory. Primary storage capacities are usually specified in number of bytes. The symbol "K" is used whenever we refer to the size of memory, especially when the memory is quite large. The symbol K is equal to 1,024 units or positions of storage. Therefore, if a computer has 512K bytes (not bits) of primary storage, then it can hold $512 \times 1,024$ or 524,288 characters (bytes) of data in its memory.

BITS AND BYTES

Computers use the binary number system using binary digits. People find that system inconvenient and prefer the octal system (which allows the digits 0,1,2...7), the hexadecimal (HEX) system (which allows the digits 0,1,2...9,A. B...F), or the familiar decimal system (which allows the digits 0,1,2...9).

EBCDIC (Extended Binary Code Decimal interchange Code)

Computer designers, first used 6 bits, instead of 4 bits with only 16 possible characters and then they moved onto 7 or 8 bits to represent characters in alphanumeric versions of BCD, with 6 bits.

Character	ASCII
0	011 0000
1	011 0001
2	011 0010
3	011 0011
4	011 0100

5	011 0101
6	011 0110
7	011 0111
A	100 0001
B	100 0010
C	100 0011
D	100 0100
E	100 0101

7-Bit ASCII format and selected ASCII character codes

Character	EBCDIC ASCII
0	1111 0000 0101 0000
1	1111 0001 0101 0001
2	11110010 0101 0010
3	1111 0011 0101 0011
4	1111 0100 0101 0100
5	1111 0101 0101 0101
6	1111 0110 0101 0110
7	1111 0111 0101 0111
A	1100 0001 1010 0001
B	1100 0010 1010 0010
C	1100 0011 1010 0011
D	1100 0100 1010 0100
E	1100 0101

1010 0101

Selected characters represented by EBCDIC and ASCII_8 bit codes

With 6 bits, it's only possible to represent 64 different characters (2^6). Since 64 bit combinations are not sufficient to provide decimal numbers (10), lowercase letters (26), capital letter (26), and a large number of other characters, designers soon extended the 6-bits BCD code to 7 and 8 bits. With 7 bits, it is possible to provide 128 different arrangements (2^7); and with 8 bits, 256 variations are possible (2^8). In addition to the four numeric place positions, there are three zone bit position in a 7 bit code and for four zone bit position in an 8-bit code. The 7-bit American standard Code for Information Interchange is widely used in data communications work and is the code used to represent data internally in personal computers. Figure shows the ASCII format and the coding used to represent selected characters.

The 8-bit codes most commonly used are:

1. Extended Binary Coded Decimal Interchange Code (EBCDIC). This is used in IBM mainframe models.
2. ASCII-8 is an 8-bit version of ASCII that is frequently used in larger machines.

PROGRAM DEVELOPMENT

1. Program definition: Understanding the program by study.
2. Program Design: Break the program into smaller modules for easy modifications.
3. Code: Select a high level language to write a program and test it.
4. Debugging: The removal of errors, correcting syntax and logical errors, and run time error is called debugging.
5. Testing: Testing is the process of making sure that the program performs intend task. It is the process of validating the correctness of a program.
6. Documentation: It is an ongoing process, which starts in study phase of the software and continues, until its implementation and operation phase.
7. Maintenance: This involves the expansion of program and change of hardware and software.

Make sure that a program is simply designed to avoid regular maintenance

Algorithm

➤ It's a step by step description of how to arrive at a solution of a given problem.

➤ It may be defined as a sequence of steps to get the desired result.

Rules for Writing Algorithm

1. Each and every instruction should be accurate.
2. Each operation should be performed in a final time.
3. After the algorithm is finished each desire result must be obtained.

Algorithm to find the greater of two numbers:

1. Start
2. Read two numbers and call them A and B respectively.
3. If A is greater than B, print A otherwise print B
4. Stop.

RUNNING AND DEBUGGING PROGRAMS

Running and debugging is the discovery and correction of programming errors.

Few programs run correctly the first time, so debugging is an important and time consuming stage in software development.

RUNNING PROGRAMS

Steps to run the program:

Step 1: Create a source code for the program.

Step 2: Save the program with its file name and extension.

Step 3: Compile and Execute the saved file.

Step 4: After execution, if any errors occur rectify the errors and repeat step 3 and 4 till correcting errors.

Step 5: Output will be displayed.

DEBUGGING

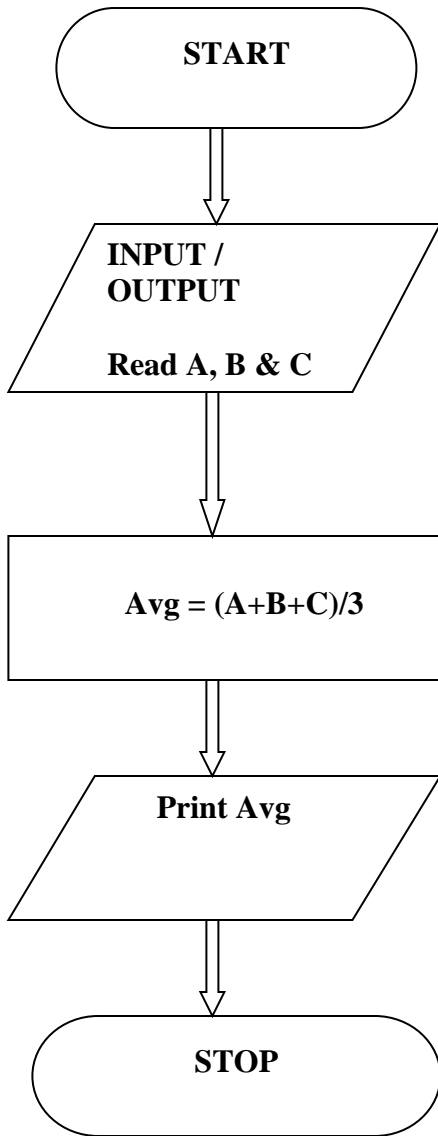
Debugging is the process of eliminating errors in the program.

- Is a reactive process.
- Can begin after the program is coded.
- Is complete when all known errors in the program have been fixed.

For example: Find the average numbers of A, B and C in algorithm.

Algorithm:

1. Start
2. Read the three values given
3. $\text{Average} = \frac{A+B+C}{3}$
4. Print Average
5. Stop.



Decision tables:

Decision tables are a precise yet compact way to model complicated logic. Decision tables, like if-then-else and switch-case statements, associate conditions with actions to perform. But, unlike the control structures found in traditional programming languages, decision tables can associate many independent conditions with several actions in an elegant way.

Decision tables are typically divided into four quadrants, as shown below.

The four quadrants	
Conditions	Condition alternatives
Actions	Action entries

Example

The limited-entry decision table is the simplest to describe. The condition alternatives are simple Boolean values, and the action entries are check-marks, representing which of the actions in a given column are to be performed.

A technical support company writes a decision table to diagnose printer problems based upon symptoms described to them over the phone from their clients.

Printer troubleshooter									
		Rules							
Conditions	Printer does not print	Y	Y	Y	Y	N	N	N	N
	A red light is flashing	Y	Y	N	N	Y	Y	N	N
	Printer is unrecognized	Y	N	Y	N	Y	N	Y	N
Actions	Check the power cable			X					
	Check the printer-computer cable	X		X					
	Ensure printer software is	X		X		X		X	

	installed								
	Check/replace ink	X	X			X	X		
	Check for paper jam		X		X				

Of course, this is just a simple example (and it does not necessarily correspond to the reality of printer troubleshooting), but even so, it demonstrates how decision tables can scale to several conditions with many possibilities.

Pseudo code:

Pseudo code is another programming analysis tool, which is used for planning program logic. “Pseudo” means imitation or false, and “Code” refers to the instructions written in a programming language. Pseudo code, therefore, is an imitation of actual computer instructions. These pseudo –instructions are phrases written in ordinary natural language (e.g., English, French, German etc.) when pseudo code is used for program planning, a programmer can concentrate solely on developing the logic of the program without worrying about the syntax for writing program instructions, because pseudo code does not have any syntax rules for formulating instructions. Pseudo code emphasizes the design of the program, it is also called Program Design Language (PDL).

Pseudo code for Basic Logic structures

1. Sequence logic
2. Selection logic, and
3. Iteration (or looping) logic.

Sequence logic: Sequence logic is used for performing instructions one after another in sequence.

Selection logic: Selection logic also known as decision logic is used for making decisions. It is used for selecting the proper path, out of two or more alternative paths, in program logic.

Iteration (or looping) Logic: Iteration logic is used to produce loops in program logic, when one or more instructions may be executed several times depending on some condition. It uses two structures called DO...WHILE and REPEAT...UNTIL.

An example of how pseudo code differs from regular code is below.

Regular code (written in PHP):

```
<?php
```

```
if (is valid($cc_number)) {  
    execute transaction($cc_number,  
$order);  
} else {  
    show failure();  
}  
?>
```

Pseudo code:

```
if credit card number is valid then  
    execute transaction based on  
    number and order  
else  
    show a generic failure message  
end if
```

Review Questions:

1. **Convert the following into Binary number:**
 - $(72)_8$
 - $(EF)_H$
 - $(98)_D$
2. **Convert the following into HEXA DECIMAL:**
 - $(1011\ 0010)$
 - $(76)_8$
3. **Explain with Examples of the following:**
 - ASCII
 - BCD
 - EBCDIC
4. **Write down the flow chart Symbols.**
5. **Write an algorithm and flow chart for greatest of three numbers.**

Chapter IV C Language

Aim:

The student is expected to fully comprehend and demonstrate an understanding on the fundamental elements of C language.

Objective:

- Acquire the basic knowledge of the building blocks in C language.
- Posses the necessary skill to write simple programs in C language

Elements of C language: Character set, keywords, identifiers, constants, variable, data types, header files and comment statements; declaring, initializing and assigning values to variables.

Operators, expressions and assignment statements: Arithmetic, relational, logical, increment, decrement, conditional, shorthand, assignment operators with hierarchy rules; evaluation of expressions and assignments statements.

I/O Statements:Printf and scanf functions (unformatted and formatted), getchar () and putchar () functions.

Functions: Predefined functions (definition and syntax only) – is digit, isalpha, isupper, is lower and ispunct functions in the header file <ctype.h>; sin, cos, tan, exp, ceil, floor, abs, pow and sqrt functions in the header file <math.h>; strlen, strcpy, strcmp and strstr in the header file <string.h>.

Control Statements:Unconditional goto, if ... else, else if ladder, nested if, switch, for loops, do, while loops and nested loops.

ELEMENTS OF C LANGUAGE:

Character set:

A Character Set is a set of valid characters that a language can recognize. It is used to represent information. The C Language has the following character set:

- **ALPHABETS** : A-Z,a-z
- **DIGITS** : 0-9
- **SPECIAL SYMBOLS** : Space + - / ^ \0 [] { } =! < > ‘ “ \$, ; ; % ! & ? _ # <= >= @
- **WHITE SPACES** : Blank Space, Horizontal tab(->),Carriage return, Newline, Formfeed

➤ **OTHER CHARACTERS** : 256 ASCII Characters as data or as literals.

Keywords:

Keywords are the words that convey a special meaning to the language compiler. These are reserved for special purpose and must not be used as normal identifier names. There are only 32 keywords available in C

Examples: Asm, continue, float, new, try, break, goto, if, int etc. auto, break, case, char, const, continue, default, do, double, else, enum, extern, float, for, goto, if, int, long, register, return, short, signed, sizeof, static, struct, switch, typedef, union, unsigned, void, volatile, while

Identifiers:

In C, the names of variables, functions, labels and various other user defined objects are called identifiers. The first character of an identifier must be a letter or an underscore and subsequent can either be a letter, digit or underscore.

Examples: Counter, Var12, P_num

Constants:

C allows several kinds of literals or Constants

➤ Integer literals

➤ Character literals

➤ Floating literals

➤ String literals

➤ **Integer literals:** An integer value which is given in the variable declaration part.

➤ **Character literals:** A value is enclosed between single a quote which is given in the variable declaration part.

➤ **Floating literals:** It is represented by the exponential notation.

Ex: The number 34578.34 would be written as 3.457834E4. ‘E’ is called exponent.

➤ **String Constants:** These are all enclosed between double quotes. **Ex:** “JESUS”

Variable (declaring, initializing and assigning values to variables):

Variable: A variable represents a place in main memory that stores a piece of data. Every variable has a name, given to it by the programmer.

In C, variable names must adhere to the following rules:

- The name can contain letters, digits, and the underscore character (_).
- The first character of the name must be a letter. The underscore is also a legal first character, but its use is not recommended.
- Case matters (that is, upper- and lowercase letters). Thus, the names count and Count refer to two different variables.
- C keywords can't be used as variable names. A keyword is a word that is part of the C language.

Local variables: Variables that are declared inside a function are called local variables. They are not known outside their own functions.

Example : main()

```
    {  
function1()  
    {  
int s; //s is a local variable  
    --  
    }  
    }
```

- Life time of a variable 's' is within function1

Global variables: Global variables are known throughout the program and may be used by any piece of code.

Example

```
main()  
{  
int a; // global variable  
    {  
f1()  
    {---  
    ---  
    }  
--
```

```
}
```

Life time of a variable 'a' is throughout the program

1. **Integer variables:** It is used to store whole numbers.

Ex:

```
int var1;          //Defines an integer variable var1
int var2=10;      //Defines and assigns value to var2
```

2. **Character Variables:**

It is used to store character constants like 's','c' which is enclosed within single quotes.

Ex:

```
char var1;        //Defines a character variable var1
char var2='g';    //Defines and assigns character value to var2
```

3. **Float variables:**

It is used to store floating point numbers like 4.34,6.7895 and 0.00456

There are three kinds of floating point variables in c

- Float
- Double
- Long double

Ex:

```
Float fvar1=20.50 //Defines a float variable and assigns a value to it
```

Declaration/Initialization of variables:

You can give variables a value as you declare them by placing an equal sign (=) and a value after the variable name.

Syntax for Declaration of variables: Type variable name;

Ex:

```
charch;
char name;
intfirst,second;
```

Syntax for Initialization of variables: Type variable name=value;

Ex:

```
char ch='a';
int first='5';
floatbal=23.456
```


Data types:

Data types are means to identify the type of data and the associated operation of handling it. Data can be of many types.

These are the:

➤ Built-in Data types:

✓ **Char** : Anything enclosed in single quotes represents Character data.

It is a data type used to store the character values such as 's', 'k'

Syntax: <data type><variables><semicolon>

Example : char a='lusaka';

✓ **int** : Numbers without fractions represents integer data

It is a data type used to declare the integer values such as 1, 2, 3...etc

Syntax: <data type><variables><semicolon>

Example : int a=5;

✓ **float** : Number with fractions represents float data

It is a data type used to store the floating point values such as 3.4, 1.5, ...etc..

It is use to occupai byte

Syntax: <data type><variables><semicolon>

Example: float a=5.5;

✓ **void** : A variable which returns nothing is void data

✓ **Double**

It is a data type used to store large floating point values such as 34.564, 45.76565...etc..

Syntax: <data type><variables><semicolon>

Example: double a=5.554;

Integer Types

Following table gives you details about standard integer types with its storage sizes and value ranges:

Type	Storage size	Value range
char	1 byte	-128 to 127 or 0 to 255
unsigned char	1 byte	0 to 255
signed char	1 byte	-128 to 127

int or Short int	2 bytes	-32,768 to 32,767 or -
Long int	4 bytes	2,147,483,648 to 2,147,483,647
unsigned int	2 bytes	0 to 65,535 or 0 to 4,294,967,295
short	2 bytes	-32,768 to 32,767
unsigned short	2 bytes	0 to 65,535
long	4 bytes	-2,147,483,648 to 2,147,483,647
unsigned long	4 bytes	0 to 4,294,967,295

Floating-Point Types

Following table gives you details about standard floating-point types with storage sizes and value ranges and their precision:

Type	Storage size	Value range	Precision
float	4 byte	1.2E-38 to 3.4E+38	6 decimal places
Double	8 byte	2.3E-308 to 1.7E+308	15 decimal places
long double	10 byte	3.4E-4932 to 1.1E+4932	19 decimal places

➤ User Defined Data types:

It is constructed from the built-in types. Ex: Arrays, Functions, Pointers etc.

Definition

- C allows us to define our own types based on other existing data types.
- In order to do that we shall use keyword typedef, whose form is:
`<typedef><existing_type><new_type_name>;`
 where ,
 existing_type is a C fundamental or any other defined type
 new_type_name is the name that the new type we are going to define will receive.

Example:

```
typedef char C;
typedef int WORD;
```

```
typedef char * string_t;
typedef char field [50];
```

Comment statements:

- Single comment statement:
One line comment statement is called as Single comment statement.
// Example program to demonstrate for statement
- Multiple comment statements:
Multiple Lines of comment statements are called as Multiple comment statement.

Header files:

The #include directive instructs the C compiler to add the contents of an include file into your program during compilation. An *include file* is a separate disk file that contains information needed by your program or the compiler. Several of these files (sometimes called header files) are supplied with your compiler. Include files should all have an .H extension (for example, STDIO.H). It deals with the function and data types in associated with input/output statements.

```
/*Example program to  
Demonstrate for statement*/
```

OPERATORS, EXPRESSIONS AND ASSIGNMENT STATEMENTS:

Operators:

All computer languages provide tools for some predefined operations. These tools are known as operators. The operator and the operands together, constitute an expression.

The various categories of operators are:

- ✓ Arithmetic operators
- ✓ Relational operators
- ✓ Logical operators
- ✓ Increment operators
- ✓ Decrement operators
- ✓ Conditional operators
- ✓ Shorthand assignment operators

Arithmetic Operators:

In general, there are four arithmetic operators +, -, *, /, C also provide the **remainder** or the **modulo** operator represented by the % symbol which returns the remainder when an integer is divide by another. **Ex:** 4+2-3

The symbols + and - are operators and constants 4, 2 and 3 are operands.

All these operators are 'binary' operators since they operate on two operands at a time.

Relational Operators:

These operators are used to compare two operands to see whether they are equal to each other, unequal, or whether one is greater than the other.

RELATIONAL OPERATORS	MEANING
>	Greater than
<	Less than
= =	Equal to
!=	Not Equal to
>=	Greater than or Equal to
<=	Less than or Equal to

Each relational operator needs two operands for comparison of their values.

Ex: a<b, d>=5, The expression a<b means "is a less than b?". The result of comparing a with b may be either "true" or "false".

Logical Operators:

It is used to combine two or more test expression. C provide the AND(&&), the OR(||) and the NOT(!) logical operators

LOGICAL OPERATOR	ACTION
!	Not (is a unary operator which means that it takes only one operand)
&&	AND (combines two conditional expressions and evaluates to true only if both conditions are fulfilled)
	OR (combines two conditional expressions, and evaluates to true if at least one of the conditions is fulfilled)

Increment Operators and Decrement Operators:

In C, there are two useful operator not generally found in other computer language. These are the Increment (++) and Decrement (--) operators. The operator ++ adds 1 to its operand and -- subtracts 1.

Ex:

++X ; is same as X=X+1

--X; is same as X=X-1;

The operator ++ and --can be used as both postfix and prefix operators

Ex:

Y=++x; is equal to y=(x+=1)

Y=x++; is equal to y=x;

As a prefix, the operator precedes the variable e.g. ++x;

As a post fix, the operator follows the variable e.g. x++;

The difference between two means, such as prefix like --i, decrements the value of i by 1 before it is used for computation, while the postfix like i--,decrements the current value of i after it has been used for calculation.

Conditional Operators:

Syntax: <exp1>? <exp2> :< exp3>;

If expression1 is true, then the value of the whole expression is the value of expression2. Otherwise, the value of the whole expression is the value of expression3.

Ex: Y= (x > 9)? 30: 40; - Y will be assigned the value 30, if X is greater than or equal to 9, then y would receive the value of 40.

Shorthand assignment operators:

OPERATOR	EFFECT
+=	Assigns Left operand+Right operand to Left operand. Ex: a+=10; means a=a+10;
-=	Assigns Left operand-Right operand to Left operand. Ex: a-=10; means a=a-10;
*=	Assigns Left operand*Right operand to Left operand. Ex: A*=10; means a=a*10;

/=	Assigns Left operand/Right operand to Left operand. Ex: a/=10; means a=a/10;
%=	Assigns Left operand% Right operand to Left operand. Ex: A%=10; means a=a%10;

Hierarchy rules:

Evaluation of expressions and assignments statements:

The C assignment statement assigns values to variables.

Syntax: Variable=expression;

Ex:

int j=2;

I=3+4*5

I/O STATEMENTS:

Printf and scanf functions (unformatted and formatted):

Scanf() function is used to input values from the keyboard

Printf() is a function which is used to print the value contained in a variable on the screen.

The Printf () function outputs formatted data on the standard output, the screen. The arguments of the function are the values to be printed. As with scanf () a control string for formatting the output is given before the arguments.

The general form of scanf function is: scanf ("format string",&list of variables);

The general form of printf function is: printf ("format string", list of variables);

format string could be:

%c for printing character values.

%d for printing integer values.

%f for printing float values.

%lf for printing double values.

getchar() and putchar() functions:

The getchar () and putchar () Function:

The getchar () function is used to receive a string from the keyboard. It is terminated when an enter key is hit. Thus space and tabs are perfectly acceptable as part of the input string.

The putchar () function works exactly opposite to getchar() function .It outputs a string to the screen. putchar() and getchar() can be used to read only one string at a time.

```
//program to demonstrate putchar() and getchar() functions
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char name[40];
clrscr();
puts("Enter your name :");
gets(name);
puts("your name is :");
puts(name);
getch();
}
```

FUNCTIONS – (Predefined functions)

Header files:

Header file <ctype.h>: is digit, isalpha, isupper, is lower and ispunct functions:

isdigit():

Returns the numeric value of the given alphabet.

Syntax: isdigit(a);

isalpha():

Returns the alphabet value of the given number.

Syntax: isalpha(25);

```
#include<stdio.h>
#include<conio.h>
#include<ctype.h>
void main()
{
char name;
clrscr();
printf("enter the numeric value");
scanf("%d",&name);
```

```
isalpha(name);
printf("the equivalence of the given value is:=%c",name);
getch();
}
```

isupper():

Returns the upper case value of the given lower case letters.

Syntax: isupper("hello");

islower():

Returns the lower case string for the given upper case string.

Syntax: islower("HAI");

Header file <math.h>:sin, cos, tan, exp, ceil, floor, abs, pow and sqrt functions:

abs():

Returns the absolute value of an integer.

Syntax: abs (int x);

fabs():

Returns the absolute value of a floating point number

Syntax: double fabs (double x)

Program to demonstrate abs() and fabs() functions

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
intn,t=1,d=1;
clrscr();
printf("Enter the values\n");
scanf("%d", &n);
t=abs(n);
d=fabs(n);
printf("Absolute value is%d\n",t);
printf("Fabsolute value is%d\n",d);
getch();
```



```
}
```

log() :

Calculates the natural logarithm of x.

Syntax: double log (double x)

log 10():

Calculates log(x) to the base 10.

Syntax: double log10 (double x);

Program to demonstrate log () and log10 () functions

```
#include <math.h>
#include <stdio.h>
#include<conio.h>
int main(void)
{
clrscr();
double result, result1;
double x = 800.6872;
result1=log(x);
result = log10(x);
printf("the log value of %f is %f\n",x,result1);
printf("The common log10 of %lf is %lf\n", x, result);
getch();
return 0;
}
```

Pow():

Calculates x to the power of y.

Syntax: double pow (double x, double y)

Program to demonstrate pow () function

```
#include <math.h>
#include <stdio.h>
#include<conio.h>
int main(void)
{
clrscr();
double x = 10.0, y = 05.0;
printf("%4.2f raised to %4.2f is %4.2f\n", x, y, pow(x, y));
getch();
}
```

```
return 0;  
}
```

Sqrt ():

Calculates the positive square root of input value.

Syntax: double sqrt(double x);

Program to demonstrate square root function

```
#include <math.h>  
#include <stdio.h>  
# include <conio.h>  
void main()  
{  
clrscr();  
double x , result;  
printf("Enter the value of x to find the square root");  
scanf("%lf",&x);  
result = sqrt(x);  
printf("\nThe square root of %lf is %lf\n", x, result);  
getch();  
}
```

Sin ():

Calculates the sine of the angle specified in radians.

Syntax : double sin (double x);

Cos ():

Calculates the cosine

Syntax: double cos (double x);

Program to demonstrate sin() and cos() functions

```
#include<stdio.h>  
#include<conio.h>  
#include<math.h>  
void main()  
{  
clrscr();  
floatx,t,s,sum;  
int i;  
printf("Enter Values in degrees\n");
```

```

scanf("%f",&x);
x=(x*3.14)/180;
printf("\nthe value of x %f",x);
t=sin(x);
s=cos(x);
printf("\nconverted sin value %f",t);
printf("\nconvertedcos value %f",s);
getch();
}

```

randomize() & random():

random();

random returns a random number between 0 and (num-1).random(num) is a macro defined in stdlib.h.

random(num); //Macro that initializes random number generator
int random(intnum);

randomize();

Remarks:

randomize initializes the random number generator with a random value.

Declaration:

void randomize (void);

Because randomize is implemented as a macro that calls the time function. Prototyped in time.h, you should include time.h when you use this routine.

Program to demonstrate random () and randomize () functions

```

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include <time.h>
/* prints a random number in the range 0 to 99 */
int main(void)
{
clrscr();
randomize();
printf("Random number in the 0-99 range: %d\n", random (100));
getch();
return 0;

```

```
}
```

Program to demonstrate random no using rand () function

```
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
int main(void)
{
clrscr();
int i;
printf("Ten random numbers from 0 to 99\n\n");
for(i=0; i<10; i++)
printf("%d\n", rand() % 100);
return 0;
}
```

Header file <string.h>:strlen, strcpy, strcmp and strCact:

Any group of characters enclosed in double quotes is a string constant.

A string is an array of characters terminated by a NULL character.

Declaring Strings:

The string that can hold an array of characters can be declared as, **char variable_name [size];**

For Example: char name[10], book [10]; - where **char** is a datatype, **name & book** is the variable and **10** is the size of the variable.

Example Programs:

AIM:

Write a Program in C to find the reverse a String.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char string[100];
clrscr();
puts("Enter the String:");
gets(string);
```

```
strrev(string);  
printf("Reversed String is:");  
puts(string);  
getch();  
}
```

RESULT: Thus the string is reversed successfully.

OUTPUT:

Enter the String: jump

Reversed string is: pmuj

AIM:

Write a Program in C to copy the string using string function

PROGRAM:

```
#include<stdio.h>  
#include<conio.h>  
#include<string.h>  
void main()  
{  
char string1[100];  
char string2[100];  
clrscr();  
puts("Enter the First String:");  
gets(string1);  
puts("Enter the Second String:");  
gets(string2);  
strcpy(string1,string2);  
printf("After copying, First string is:");  
puts(string1);  
getch();  
}
```

RESULT: Thus the string is copied successfully.

OUTPUT:

Enter the First String: Zambia

Enter the Second String: Lusaka

After copying, First string is: Lusaka

AIM:

Write a Program in C to concatenate two strings using string function.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char string1[100];
char string2[100];
clrscr();
puts("Enter the First String:");
gets(string1);
puts("Enter the Second String:");
gets(string2);
strcat(string1,string2);
printf("After concatenation the string is:");
puts(string1);
getch();
}
```

RESULT: Thus the string is concatenated successfully.

OUTPUT:

Enter the First String: welcome

Enter the Second String: to zambia

After Concatenation the string is: welcome to zambia

Control Statements:**Unconditional goto:**

if ... else, else if ladder, nested if:

Simple if:

Explanation: When the condition is true, the following action is performed.

Syntax:

If (test condition)

Statement

Example:

```
If (i>9)
Printf (“welcome”);
```

If.....Else:

Explanation: When the condition is true, the following action is performed and certain other action when false.

Syntax:

```
if (test condition)
statement;
else
```

```
    Statement;
```

Example:

/*a simple c program to find the greatest of two numbers, example to demonstrate IF statement*/

```
#include <stdio.h>
#include<conio.h>
void main ( )
{
clrscr( );
inta,b;
printf(enter two numbers”);
scanf(“%d%d”,&a,&b);
if (a>b)
printf(“a is greater);
else
printf(“b is greater”);
getch();
}
```

Nested if:

If statement is within another if statement is called nested if.

Ex:

```
If (a>b)
    If (a>c)
        Big=a;
    Else
        If (b>c)
```

```

        Big=b;
    Else
        Big=c;
/* a simple c program to find the greatest of three numbers*/
/*an example to demonstrate IF – else statement*/
#include <stdio.h>
#include<conio.h>
void main ( )
{
clrscr( );
inta,b,c;
printf(“enter three numbers”);
scanf(“%d%d%d”,&a,&b,&c);
if((a>b)&&(a>c))
printf(“ a is greater”);
else if((b>c)&&(b>a))
printf(“b is greater”);
else
printf(“ c is greater”);
getch();
}

```

Switch:

Multiple branching is achieved with the help of switch statement as follows. The expression should result is an integer (character also) value. Control is transferred to the case statement whose label matches this value. Case labels have to be constants .The case label is followed by a colon and the statements that are to be executed under this condition .This set of statements usually end with a break statement this causes control to jump out of the switch structure. A default case may also be included as shown below.

Syntax for switch statement:

```

Switch (expression)
{
Case 1: statements; break;
Case 2: statements; break;
.....

```


.....
Case n: statements; break;
default:
}

Ex:
#include <stdio.h>
#include <conio.h>
void main ()
{
clrscr();
int day;
printf("enter the day");
scanf("%d",&day);
switch(day)
{
case 1: printf("Monday");
break;
case 2: printf("Tuesday");
break;
case 3: printf("Wednesday");
break;
case 4: printf("Thursday");
break;
case 5: printf("Friday");
break;
case 6: printf("Saturday");
break;
case 7: printf("Sunday");
break;
default: printf("not valid");
break;
}
getch();
}

For loops:

Explanation: <statement> is executed repeatedly UNTIL the value of <expr2> is false.

BEFORE the first iteration, <expr1> is evaluated. This is usually used to initialize variables for the loop.

AFTER each iteration of the loop, <expr3> is evaluated. This is usually used to increment a loop counter.

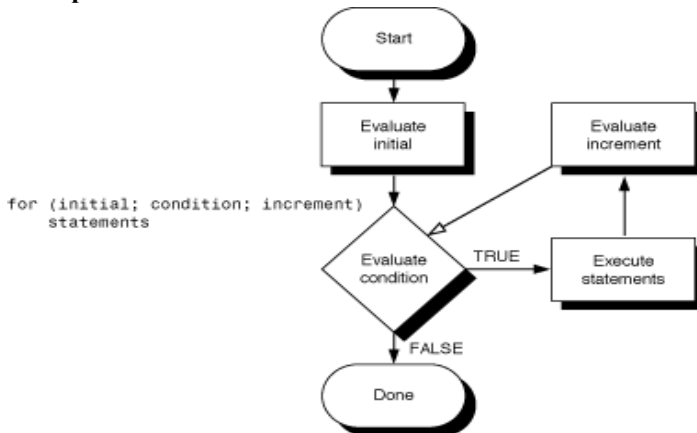
In C, <expr1> can be an expression or a declaration.

All the expressions are optional. If <expr2> is left out, it is assumed to be 1.

Syntax:

for ([<expr1>] ; [<expr2>] ; [<expr3>]) <statement>

The operation of do.... While statement:



Examples:

```
for (i=0; i<100; i++)  
    Sum += x[i];
```

Nested Loop:

Definition: A Loop within another loop is called Nested Loop.

Example: A j loop inside of another i loop is called nested loop.
for(i=0;i<10;i++)

```
for(j=0;j<10;j++)
scanf("%d",&a[i][j]);
```

/*a simple c program to find the first five Fibonacci numbers*/

/*Example program to demonstrate for statement*/

```
#include <stdio.h>
#include<conio.h>
void main ( )
{
clrscr ( );
int sum, a=1,b=1;
inti,n;
printf("\n enter the number of terms to be printed");
scanf("%d",&n);
printf("%d%d",a,b);
for(i=3,i<=n;i++)
{
sum=a+b;
printf("%d",sum);
a=b;
b=sum;
}
getch ();
}
```

do, while loops and nested loops:

while loops:

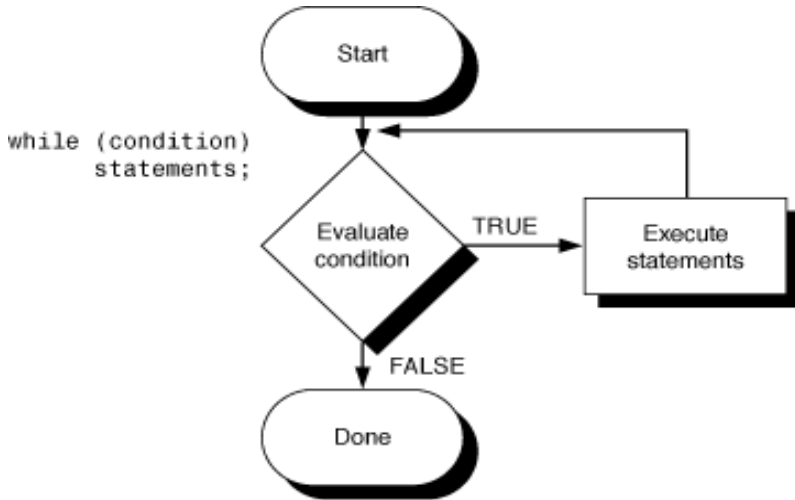
Explanation: <Statement> is executed repeatedly as long as, the value of <expression> remains true.

Syntax:

While (<expression>)

<Statement>

The operation of while statement:



Example:

```

i=1; s=0;
While (i<=10)
{
  s=s+i;
  i++;
}

```

Do while loops:

Explanation: <statement> is executed repeatedly as long as the value of <expression> remains true. The test takes place AFTER each execution of the <statement>.

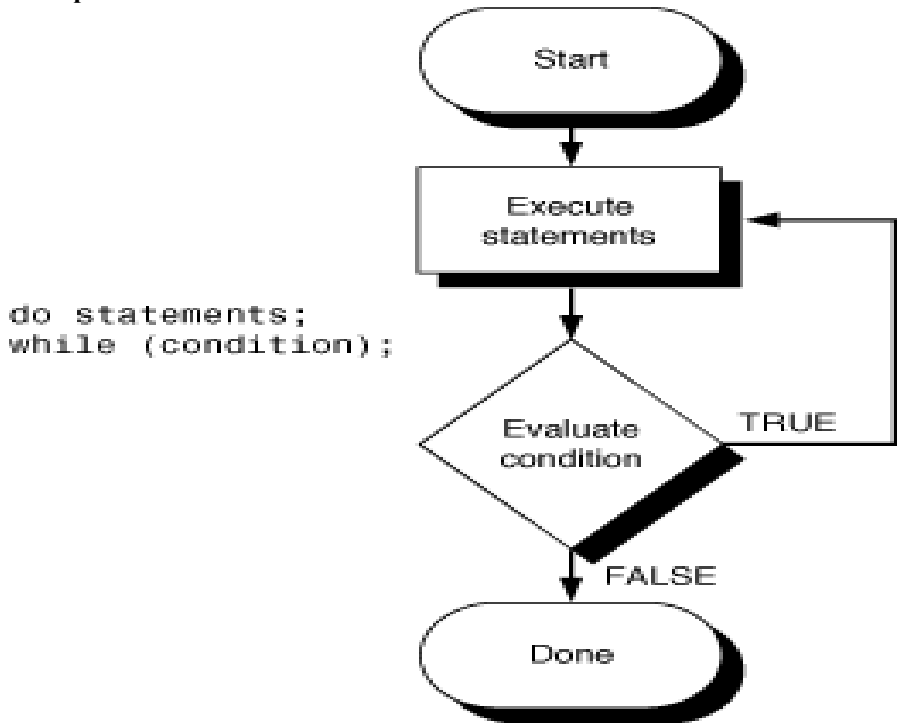
Syntax:

```

do
<statement>
while ( <expression> );

```

The operation of do.... While statement:



Example:

```
i = 1; n = 1;  
do {  
s+= i;  
i++;  
} while (i <= 10);
```

Note: The difference between while and do-while is, the while tests the condition before executing any of the statements within the while loop. As against this the do-while tests the condition after having executed the statements within the loop. This means that do-while would execute its statements at least once, even if the condition fails for the first time itself.

Review Questions:

1. Describe about elements of “C” language.
2. Summarize the statements.
3. Explain with examples about pre- defined and user defined function.
4. Write a program in “C” to find the cube of particular number by using user defined function.
5. List down some header file which is used in “ C” language.

Chapter V ARRAYS

Aim:

The student is required to possess the ability to understand arrays and their use in C programming.

Objectives:

- Demonstrate effective understanding on the definition and processing of arrays.
- Differentiate on how to apply single and two dimensional arrays.
- Identify common symptoms and problems associated with writing programs.

UNIT V:

Arrays: Defining an Array, Processing Array and Passing Arrays to Functions, Declaring, Initializing and using single, two dimensional arrays.

Arrays

An array is defined as a set of homogeneous data items of the same type that share a common name.

Declaration of One Dimensional Array

Like ordinary variables, an array variable should be declared.

GENERAL FORM OF DECLARING AN ARRAY IS :

Datatype array-Name [subscript];

Note : Datatype specifies the type of element that will be stored in the array.

array-name is a valid C variable of type datatype,

Subscript is an integer constant indicating the maximum number of elements that can be stored in the array.

For Example :

```
int a[5];
```

a [0]	a [1]	a [2]	a [3]	a [4]
-------	-------	-------	-------	-------

Important points to be noted while using arrays:

1. An array should be a valid C identifier.

2. The name of the array should be unique, similar to other variables.
3. The value of the elements stored in the array should be of the same type.

Example Programs:

Example 1:

Write a Program in C to find the sum of elements in one dimensional array.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    inta[10],i,n,sum=0; //Declaration of array variable
    clrscr();
    printf("Enter the Array size :");
    scanf("%d",&n);
    printf("Enter the values :");
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
        sum=sum+a[i];
    }
    printf("The sum of elements are :%d",sum);
    getch();
}
```

RESULT: Thus all array elements are added successfully.

OUTPUT:

```
Enter the Array size: 4
    Enter the Values: 2 5 4 2
The Sum of elements is: 13
```

Example 2:

Write a Program in C to find the product of elements in one dimensional array

```
#include<stdio.h>
```



```

#include<conio.h>
void main()
{
int a[10],i,n,mul=1;
clrscr();
printf("Enter the Array size :");
scanf("%d",&n);
printf("Enter the values :");
for(i=0;i<n;i++)
{
scanf("%d",&a[i]);
mul=mul*a[i];
}
printf("The product of elements are :%d",mul);
getch();
}

```

RESULT: Thus all array elements are multiplied successfully.

OUTPUT:

Enter the Array size: 4

Enter the Values: 2 2 2 2

The Sum of elements is: 16

Example :3

Write a Program in C to find the average of elements in one dimensional array.

```

#include<stdio.h>
#include<conio.h>
void main()
{
int a[10],i,n,sum=0;
clrscr();
printf("Enter the Array size :");
scanf("%d",&n);
printf("Enter the values :");
for(i=0;i<n;i++)
{

```

```
scanf("%d",&a[i]);
sum=sum+a[i];
    }
avg=sum/n;
printf("The Sum of elements is: %d",sum);
printf("The Average of array elements is: %d",avg);
getch();
    }
```

RESULT: Thus average of array elements is calculated successfully.

OUTPUT:

Enter the Array size: 4

Enter the Values: 2 2 2 2

The Sum of elements is: 16

The Average of array element is: 4

Initializing one-dimensional arrays:

Initialization of all elements in an array can be done at the time of declaration and definition, just as with variables.

Array can be initialized by two ways:

1. Initializing an array during declaration.
2. Initializing an array using loops.

Initializing an array during declaration:

For each element in the array, we provide a value. The only difference is that the values must be enclosed in braces and, if there is more than one, separated by commas.

General form:

Datatype array-name[size] = {list of values};

For Example 01:

Memory allocated for the array age as shown:

int age [5] = { 3 , 7, 12, 24, 45};

Array elements: age [0] age [1] age [2] age [3]

age [4]

Values:

3	7	12	24	45
---	---	----	----	----

For Example 02:

```
int age [5] = { 3, 7 };
```

Array elements: age [0] age [1] age [2] age [3]
age [4]

Values:

3	7	0	0	0
---	---	---	---	---

(The rest are filled with 0's)

Example 1:

Program to find the maximum element in a one dimensional array

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
clrscr();
inti,n,a[10], max=a[0];

printf("\n Enter the size of the array : ");
scanf("%d",&n);
printf("\n Enter the array elements");
for(i=0;i<n;i++)
scanf("%d",&a[i]);

for(i=0;i<n;i++)
{
if(a[i]>max)
max=a[i];
}
printf("\n Maximum is=%d",max);
getch();
}
```

OUTPUT:

Enter the size of the array: 3
Enter the array elements: 7 2 5

Maximum is = 7

Program 2:

Program to find the minimum element in a single dimensional array

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
clrscr();
inti,n,a[10],min=a[0];

printf("\n Enter the size of the array: ");
scanf("%d",&n);

printf("\n Enter the array elements");
for(i=0;i<n;i++)
scanf("%d",&a[i]);

for(i=0;i<n;i++)
{
if(a[i]<min)
min=a[i];
}

printf("\n Minimum is=%d",min);
}
```

OUTPUT:

Enter the size of the array: 3
Enter the array elements: 7 2 5
Minimum is = 2

Initializing array using Loops

An array can be initialized by using a for loop, while loop or do-while loop. The following example illustrate the use of initializing an array using loop :

Example :

```
int i , mark[50];
```

```

for (i=0;i<50;i++)
{
    Mark[i]=0;
}

```

Declaration of Two-Dimensional Array

A two-dimensional array is defined in the same manner as a single dimensional array except that it requires two pairs of square brackets for two subscripts. The two dimensional array is also referred as a matrix. Each subscript must be expressed as a non-negative integer.

The general form:

Datatype array-name [row-size] [column-size];

Note : Datatype specifies the type of element that will be stored in the array,

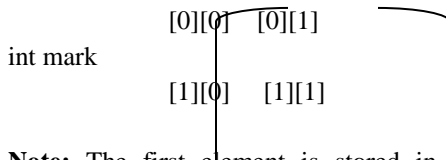
array-name is a valid C variable of type datatype,

row-size and column-size are positive integer value.

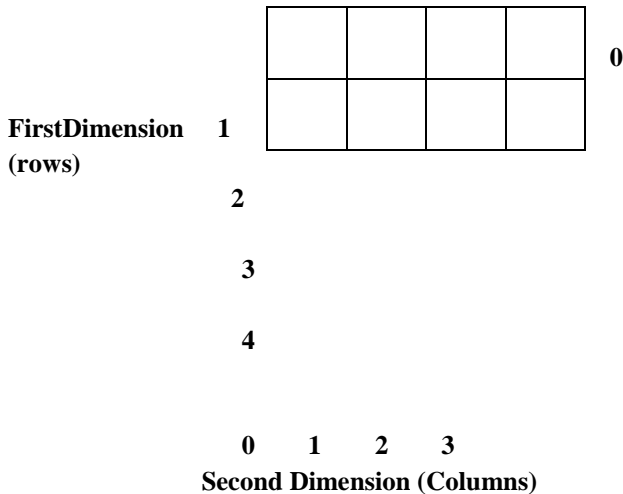
For Example :

int mark[2][2];

An array element starts with an index zero and so the individual elements of the array will be



Note: The first element is stored in the memory location [0][0], followed by the second element in [0][1] and so on, and the last element will be stored in the memory location [1][1].



Example 1:

Write a Program in C to find the Transpose matrix using two dimensional arrays.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a[10][10],i,j,m,n;
clrscr();
printf("Enter the Row size :");
scanf("%d",&m);
printf("Enter the column size :");
scanf("%d",&n);
printf("Enter the values :");
for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
scanf("%d",&a[i][j]);
}
printf("The Transpose matrix is :");
for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
```

```

        {
printf(“%d \t”,a[j][i]);
        }
printf(“\n”);
    }
getch();
}

```

RESULT: Thus Transpose matrix is found successfully.

OUTPUT:

Enter the Row size: 2

Enter the Column Size: 2

Enter the Values: 1 5 7 9

The Transpose matrix is:

```

1 7
5 9

```

Initializing two-dimensional arrays

Initialization of all elements in an array can be done at the time of declaration and definition, just as with variables.

Array can be initialized by two ways :

1. Initializing an array during declaration.
2. Initializing an array using loops.

Initializing an array during declaration:

For each element in the array, we provide a value. The only difference is that the values must be enclosed in braces and, if there is more than one, separated by commas.

General form :

Datatype array-name[row-size] [column-size] = {list of values};

For Example 01:

Memory allocated for the array mark as shown:

```

int mark [4] [2] = {
                {85, 56},
                {92, 67},
                {78, 75},

```

```
        {69, 89}
```

```
    };
```

The results of the initial assignments are,

```
mark[0][0]=85    mark[0][1]=56
mark[1][0]=92    mark[1][1]=67
mark[2][0]=78    mark[2][1]=75
mark[3][0]=69    mark[3][1]=89
```

Initializing array using Loops

An array can be initialized by using a for loop, while loop or do-while loop. The following example illustrates the use of initializing an array using loop:

Example:

```
int i , j, mark[10][5];
for (i=0;i<10;i++)
for (j=0;j<5;j++)
{
    Mark[i][j]=0;
}
```

Inputting Array Elements

Array table is initialized and inputted as shown:

```
int table [5] [4] =
{
    {0, 1, 2, 3},
    {10, 11, 12, 13},
    {20, 21, 22, 23},
    {30, 31, 32, 33},
    {40, 41, 42, 43}
}
for (row = 0; row < 5; row++)
    for (column = 0; column <4; column++)
        Scanf( "%d", &table [row] [column]);
```

When the program runs, we enter the 20 values for the elements and they are stored in the appropriate locations.

Accessing Array Elements

```
for (row = 0; row < 5 ; row++)  
    for(column = 0; column <4; column++)  
        printf ( "%d", &table [row] [column]);
```

When the program runs, we read the 20 values stored in the array and they are displayed on the screen.

Example 2:

A sample program to input and access array elements in a two dimensional matrix

```
#include<stdio.h>  
#include<math.h>  
#include<conio.h>  
void main()  
{  
clrscr();  
int a[10][10],i,j,m,n;  
printf("Enter the no of rows in the matrix");  
scanf("%d",&m)  
printf("Enter the no of columns in the matrix");  
scanf("%d",&n);  
for(i=0;i<m;i++)  
for(j=0;j<n;j++)  
scanf("%d",&a[i][j]);  
printf("The matrix is ");  
for(i=0;i<m;i++)  
for(j=0;j<n;j++)  
printf("\n%d\t\n",a[i][j]);  
getch();  
}
```

Output :

```
Enter the no of rows in the matrix: 2  
Enter the no of columns in the matrix: 2  
3 4 7 2  
The matrix is  
3 4  
7 2
```

Example 3:**PROGRAM TO ADD TWO MATRICES**

```
#include<stdio.h>
#include<math.h>
#include<conio.h>
void main()
{
clrscr();
int                                a[10][10],b[10][10],c[10][10],i,j,m,n,p,q;
//Declaration of variables
printf("Enter A matrix row and column value :\n");           //INPUT
for A matrix
scanf("%d%d",&m,&n);
printf("input A matrices\n");
for(i=0;i<n;i++)
for(j=0;j<m;j++)
scanf("%d",&a[i][j]);

printf("Enter B matrix row and column value :\\n"); //INPUT for B
matrix
scanf("%d%d",&p,&q);
printf("input B matrices\n");
for(i=0;i<p;i++)
for(j=0;j<q;j++)
scanf("%d",&b[i][j]);
for(i=0;i<n;i++)
//ADDING Matrix
for(j=0;j<m;j++)
c[i][j]=a[i][j]+b[i][j];

printf("\n sum of A and B matrices");           //OUTPUT Matrix
for(i=0;i<n;i++)
for(j=0;j<m;j++)
printf("\t %d",c[i][j]);
getch();
}
```

Output :

Enter A matrix row and column value : 2 2

```

1 2
2 3
Enter B matrix row and column value : 2 2
4 2
2 2
Sum of A and B matrix
5 4
4 5

```

Example 4:

Program to find the difference of two matrices

```

#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
int a[10][10],b[10][10],c[10][10],i,j,m,n,p,q;//Declaration of
variables
printf("Enter A matrix row and column value :\n"); //INPUT for
matrix
scanf("%d%d",&m,&n);
printf("input A matrices\n");
for(i=0;i<n;i++)
for(j=0;j<m;j++)
scanf("%d",&a[i][j]);
printf("Enter B matrix row and column value :\n"); //INPUT for B
Matrix
scanf("%d%d",&p,&q);

printf("input B matrices\n");
for(i=0;i<p;i++)
for(j=0;j<q;j++)
scanf("%d",&b[i][j]);
for(i=0;i<n;i++) //Subtraction
Matrix
for(j=0;j<m;j++)
c[i][j]=a[i][j]-b[i][j];

```

```
printf("\n difference of A and B matrices");           //OUTPUT
Matrix
for(i=0;i<n;i++)
for(j=0;j<m;j++)
printf("\t %d",c[i][j]);
getch();
}
```

Output :

```
Enter A matrix row and column value : 2 2
  4  6
  2  5
Enter B matrix row and column value : 2 2
  2  3
  1  2
difference of A and B matrix
  2  3
  1  3
```

EXAMPLE 5:

Interchanging row and column elements in a two dimensional array

Program to find the transpose of a matrix

```
#include<stdio.h>
#include<math.h>
#include<conio.h>
void main()
{
clrscr();
int a[10][10],m,n;
printf("\nEnter the no of rows & column of the matrix");
scanf("%d%d",&m,&n);
for(int i=0;i<m;i++)
for(int j=0;j<n;j++)
{
printf("\n Enter the value of the A matrix[ %d %d ]=",i,j);
scanf("%d",&a[i][j]);
}
printf("\n Transpose matrix");
```

```

for(i=0;i<m;i++)
for(j=0;j<n;j++)
printf("\n%d",a[j][i]);
getch();
}

```

Output :

```

Enter the no of rows of the matrix : 2
Enter the no of columns of the matrix : 2
Enter the value of the A matrix[ 00 ] = 2
Enter the value of the A matrix[ 01 ] = 1
Enter the value of the A matrix[ 10 ] = 6
Enter the value of the A matrix[ 11 ] = 5
Transpose matrix:
      2  6
      1  5

```

Functions

Function is a group of statements that together perform a task. Every C program has at least one function, which is **main()**, and all the most trivial programs can define additional functions.

You can divide up your code into separate functions. How you divide up your code among different functions is up to you, but logically the division usually is so each function performs a specific task.

A function **declaration** tells the compiler about a function's name, return type, and parameters. A function **definition** provides the actual body of the function.

The C standard library provides numerous built-in functions that your program can call. For example, function **strcat()** to concatenate two strings, function **memcpy()** to copy one memory location to another location and many more functions.

A function is known with various names like a method or a sub-routine or a procedure, etc.

Defining a Function

The general form of a **function definition** in C programming language is as follows:

```

return_type function_name( parameter list )
{

```

body of the function
}

A function definition in C programming language consists of a function header and a function body. Here are all the parts of a function:

Return Type: A function may return a value. The **return_type** is the data type of the value the function returns. Some functions perform the desired operations without returning a value. In this case, the **return_type** is the keyword **void**. **Parameters:** A parameter is like a placeholder. When a function is invoked, you pass a value to the parameter. This value is referred to as actual parameter or argument. The parameter list refers to the type, order, and number of the parameters of a function. Parameters are optional; that is, a function may contain no parameters.

- **Function Body:** The function body contains a collection of statements that define what the function does.

Example

```
/* function returning the max between two numbers */  
int max(int num1, int num2)  
{  
/* local variable declaration */  
int result;  
if (num1 > num2)  
result = num1;  
else  
result = num2;  
return result;  
}
```

The source code for a function called `max()`. This function takes two parameters `num1` and `num2` and returns the maximum between the two:

Function Declarations

A function declaration tells the compiler about a function name and how to call the function. The actual body of the function can be defined separately.

A function declaration has the following parts:

```
return_type function_name( parameter list );
```

For the above defined function max(), following is the function declaration:

```
int max(int num1, int num2);
```

Parameter names are not important in function declaration only their type is required, so following is also valid declaration:

```
int max(int, int);
```

Function declaration is required when you define a function in one source file and you call that function in another file. In such case you should declare the function at the top of the file calling the function.

Calling a Function

While creating a C function, you give a definition of what the function has to do. To use a function, you will have to call that function to perform the defined task. When a program calls a function, program control is transferred to the called function. A called function performs defined task, and when its return statement is executed or when its function-ending closing brace is reached, it returns program control back to the main program.

To call a function, you simply need to pass the required parameters along with function name, and if function returns a value, then you can store returned value. For example:

```
#include <stdio.h>
/* function declaration */
int max(int num1, int num2);
int main ()
{
/* local variable definition */
int a = 100;
int b = 200;
int ret;
/* calling a function to get max value */
ret = max(a, b);
printf( "Max value is : %d\n", ret );
return 0;
}
/* function returning the max between two numbers */
int max(int num1, int num2)
{
/* local variable declaration */
```

```

int result;
if (num1 > num2)
result = num1;
else
result = num2;
return result;
}

```

I kept max() function along with main() function and compiled the source code. While running final executable, it would produce the following result:

Max value is : 200

Function Arguments

If a function is to use arguments, it must declare variables that accept the values of the arguments. These variables are called the formal parameters of the function.

The formal parameters behave like other local variables inside the function and are created upon entry into the function and destroyed upon exit.

Passing Arrays to Functions

In 'C' language there are two ways, that the parameters can be passed to a function, they are

- i) Call by value
- ii) Call by reference

i) Call by value:

This method copies the values of actual parameters into the formal parameters of the function. Here, the changes of the formal parameters cannot affect the actual parameters.

Example : Find the cube of given value.

```

#include<stdio.h>
#include<conio.h>
int cube(int x)
main()
{
int n=5;
clrscr();
printf("Cube of %d is.. %d",n,cube(n));
}
int cube(int)

```



```

{
    x=x*x*x;
return(x);
}

```

OUTPUT:

Cube of 5 is.. 125

In this example, the value of argument to the function cube(), 5 is copied to the parameter 'x'. The expression x=x*x*x is evaluated, and only the local variable 'x' is modified.

ii) Call by reference:

Call by reference is another way of passing parameters to a function. Here, the address of arguments are copied into the parameter inside the function, the address is used to access the actual arguments used in the call.

Here pointers are passed to function, just like any other arguments. Here, we need not declare the parameters as pointer type.

Example : Interchanging two values.

```

#include<stdio.h>
#include<conio.h>
main()
{
int i=5,j=10;
clrscr();
printf("i and j values before:%d%d\n",i,j);
interchange(&i,&j);
printf("I and j values after:%d%d\n",i,j);
}
voidinterchange(int *a,int *b)
{
int t;

```

```
t=*a;  
*a=*b;  
*b=t;  
}
```

In this example, the variables 'i' and 'j' are assigned to values 5 and 10 respectively, then the interchange() function is called with the address of 'i' and 'j' and it interchanges the 'i' and 'j' values.

Review Questions:

- 1. Write a program in “C” to find the sum of square of series.**
- 2. Write a program in “C” to find the sum of cube of series.**
- 3. Summarize single and two dimensional arrays.**
- 4. Write a program in “C” to add two matrices.**
- 5. Write a program in “C” to transpose of particular matrix.**

References:

1. PayalLotia, Pradeep Nair, 2003, Computer and Introduction, BPB Publication, New Delhi.
2. GhoshDastidar, Chattopadhyay and Sarkar, 1999, Computers and Computation – A Beginner Guide, Prentice Hall, Delhi.
3. Robert H. Blissmer, 2000, Introducing Computers, John Wiley and Son, New York.
4. Donald H. Sanders, 2000, Computers Today, McGraw Hill International, New York.
5. Maureen Sprankle, 1998, “Problem solving and programming Concept”, Merrill Publications, Columbus.
6. D.P.Nagpal, 2000, Mastering Microsoft Office , Wheelers Publishing, New Delhi.
7. Robert H. Blissmer, 1998, “Introducing Computers”, John Wiley & Sons.
8. YashavantKanethkar, 1998, “Working with C”, BPB Publications, New Delhi.
9. Byron Gottfried, 1998, “Programming with C”, Tata Mcgraw Hill, New Delhi.