MODULE 1

Introduction to C: Introduction to computers, input and output devices, designing efficient programs. Introduction to C, Structure of C program, Files used in a C program, Compilers, Compiling and executing C programs, variables, constants, Input/output statements in C,

Textbook: Chapter 1.1-1.9, 2.1-2.2, 8.1 - 8.6, 9.1-9.14

Chapter 1: Introduction to Computers

1.1 COMPUTER

- A computer can be defined as an electronic device that is designed to accept data, perform the required mathematical and logical operations at high speed, and output the result.
- We all have seen computers in our homes, schools, and colleges.
- In the past, computers were extremely large in size and often required an entire room for installation. These computers consumed enormous amounts of power and were too expensive to be used for commercial applications.
- These days, computers have become so prevalent in the market that all interactive devices such as cellular phones, global positioning system (GPS) units, portable organizers, automated teller machines (ATMs), and gas pumps, work with computers.

1.2 CHARACTERISTICS OF COMPUTERS

A computer accepts data, processes it, and produces information.

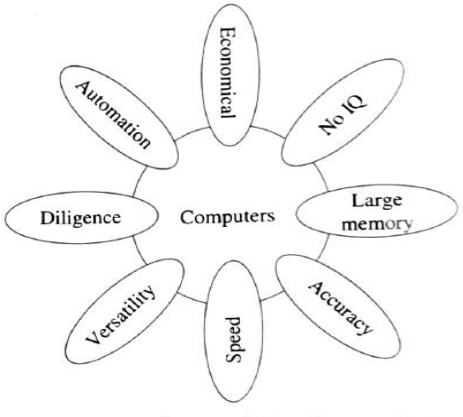


Figure 1.1 Characteristics of computers

Speed Computers can perform millions of operations per second, which means that data that may otherwise take many hours to process is output as information in the blink of an eye. The speed of computers is usually given in nanoseconds and picoseconds.

1 nanosecond = 1 x 10^{-9} seconds and 1 picosecond = 1 x 10^{-12} seconds.

Accuracy A computer is a very fast, reliable, and robust electronic device. It always gives accurate results provided the correct data and set of instructions are input to it. Hence, in the event of an error, it is the user who has fed the incorrect data/program is responsible. If the input data is wrong, then the output will also be erroneous. In computer terminology, this is known as garbage-in, garbage-out (GIGO).

Automation Besides being very fast and accurate, computers are automatable devices that can perform a task without any user intervention. The user just needs to assign the task to the computer, after which it automatically controls different devices attached to it and executes the program instructions.

Diligence Unlike humans, computers never get tired of a repetitive task. It can continually work for hours without creating errors. Even if a large number of executions need to be executed, each and every execution requires the same duration, and is executed with the same accuracy,

Versatile Versatility is the quality of being flexible. Today, computers are used in our daily life in different fields. For example, they are used as personal computers (PCs) for home use, for business-oriented tasks, weather forecasting, space exploration, teaching, railways, banking, medicine, and so on.

On the PC that we use at home - we may play a game, compose and send e-mails, listen to music, etc. Therefore, computers are versatile devices as they can perform multiple tasks of different nature at the same time,

Memory Similar to humans, computers also have memory. Just the way we cannot store everything in our memory and need secondary media, such as a notebook, to record certain important things, computers also have internal or primary memory (storage space) as well as external or secondary memory. While the internal memory of computers is very expensive and limited in size, the secondary storage is cheaper and of bigger capacity.

Some examples of secondary devices include floppy disks, optical disks (CDs and DVDs), hard disk drives (HDDs) and pen drives.

When data and programs have to be used, they are copied from the secondary memory into the internal memory, often known as random access memory (RAM).

No IQ Although the trend today is to make computers intelligent by inducing artificial intelligence (AI) in them, they still do not have any decision-making abilities of their own. They need guidance to perform various tasks.

Economical Today, computers are considered as short-term investments for achieving long-term gains. Computers save time, energy and money. When compared to other systems, computers can

do more work in lesser time. For example, using the conventional postal system to send an important document takes at least two to three days, whereas the same information when sent using the Internet (e-mail) will be delivered instantaneously.

1.3 STORED PROGRAM CONCEPT

All digital computers are based on the principle of stored program concept which was introduced by Sir John Von Neumann in the late 1940s.

The following are the characteristic features of this concept:

- Before any data is processed, instructions are read into memory.
- Instructions are stored in the computer's memory for execution.
- Instructions are stored in binary forms (using binary numbers only 0s and 1s).
- Processing starts with the first instruction in the program, which is copied into a control unit circuit. The control unit executes the instructions.
- Instructions written by the users are performed sequentially until there is a break in the current flow.
- Input / Output and processing operations are performed simultaneously. While data is being read/written, the central processing unit (CPU) executes another program in the memory that is ready for execution.

A stored program architecture is a fundamental computer architecture where-in the computer executes the instructions that are stored in its memory.

Today, a CPU chip can handle billions of instructions per second.

1.3.1 Types of Stored Program Computers

A computer with a Von Neumann architecture stores data and instructions in the same memory. There is a serial machine in which data and instructions are selected one at a time. Data and instructions are transferred to and from memory through a shared data bus. Since there is a single bus to carry data and instructions, process execution becomes slower.

Later **Harvard University** proposed a stored program concept in which there was a separate memory to store data and instructions. Instructions are selected serially from the instruction memory and executed in the processor. When an instruction needs data, it is selected from the data memory. Since there are separate memories, execution becomes faster.

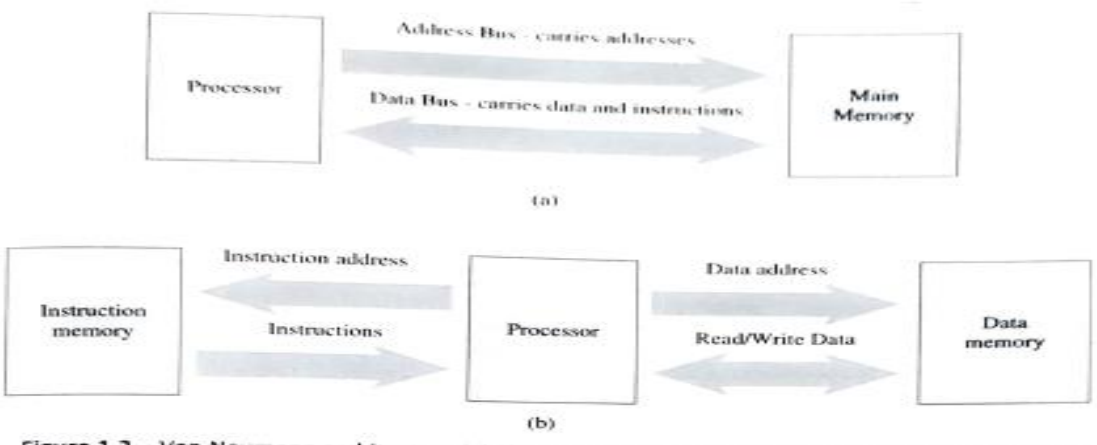


Figure 1.2 Von Neumann architecture (a) Shared memory for instructions and data (b) Separate memories for instructions and data

1.4 HISTORY OF COMPUTERS

Early computers were designed not for entertainment but for solving number-crunching problems. These computers were punch-card based computers that took up entire rooms.

Today, our smartphones have much more computing power than that was available in those early computers.

Timeline of Developments

300 BC: The abacus was an early aid for mathematical computations and was designed to aid human's memory while performing calculations. A skilled abacus operator can add and subtract with the same speed as that of a person performing the same calculation using a hand calculator.

1822: English mathematician Charles Babbage designed a steam-driven calculating machine that could compute tables of numbers. Though the project failed as he could not complete the construction of the engine, it laid the foundation for the first computer.

1890: Herman Hollerith, an American inventor, designed a punched card system to calculate the 1880 census. The system completed the task in three years saving the US government S5 million. Later Herman established a company that we today know as IBM.

1936: British mathematician Alan Turing introduced a universal machine called the Turing machine capable of computing anything that is computable. The central concept of the modern computer is based on this machine.

1941: John Vincent Atanasoff, a Bulgarian-American physicist, and his graduate student, Clifford Berry, at Iowa State College designed Atanasoff-Berry computer (ABC) that could solve 29 equations simultaneously. It was the first time a computer could store information in its main memory.

1943-1944: John W. Mauchly and J. Presper Eckert built the Electronic Numerical Integrator and Calculator (ENIAC), which is considered as the grandfather of digital computers. It filled a 20 x 40 feet room and had 18,000 vacuum tubes.

1946: Mauchly and Presper designed the UNIVAC, which was the first commercial computer for

business and government applications.

1947: William Shockley, John Bardeen and Walter Brattain of Bell Laboratories invented the transistor. Vaccuum tubes in computers were replaced by transistors.

1948: Grace Hopper developed the first computer language COBOL

1953: Grace Hopper developed the first computer language COBOL

1954: The FORTRAN programming language was developed.

1958: Jack Kilby of Texas Instruments and Robert Noyce at Fairchild Semiconductor corporation separately invented integrated circuit, which is commonly known as the computer chip.

1964: Douglas Engelbart developed a prototype of the modern computer, with a mouse and a graphical user interface (GUI). This was a remarkable achievement as it shifted computers from a specialized machine for scientists and mathematicians to general public

1969: Unix operating system was developed at Bell Labs, It was written in the C programming language and was designed to be portable across multiple platforms.

1970: DRAM chip was introduced by Intel

1971: Alan Shugart with his team in IBM invented the floppy disk which allowed data to be shared among computers.

1973: Robert Metcalfe, a research member at Xerox, developed Ethernet for connecting multiple computers and other hardware.

1974-1977: Personal computers started becoming popular.

1975: Paul Allen and Bill Gates started writing software for the Altair 8800 using the new BASIC Language. On April 4, they both formed their own software company, Microsoft.

1976: Steve Jobs and Steve Wozniak started Apple Computers and developed Apple 1, the first computer with a single-circuit board

1977: Apple II was launched that offered colour graphics and incorporated an audio cassette drive for storage.

1978: WordStar, a word processor application, was released by MicroPro International.

1979: VisiCalc, the first computerized spreadsheet program for personal computers, was unveiled.

1981: The first IBM personal computer was introduced that used Microsoft's MS-DOS operating system. The term PC was popularized.

1983: The first laptop was introduced. Moreover, Apple introduced Lisa as the first personal computer with a GUI with drop-down menus and icons

1985: Microsoft announced Windows as a new operating system.

1986 Compaq introduced Deskpro 386 in the market which was a 32 bit architecture machine that provides speed comparable to mainframes.

1990 Tim Berners-Lee invented World Wide Web with HTML as its publishing language.

1993: The pentium microprocessor introduced the use of graphics and music on PCS

1994: PC games became popular.

1996: Sergey Brin and Larry Page developed the Google search engine at Stanford University.

1999 The term Wi-Fi was introduced when users started connecting to the Internet without wires.

2001: Apple introduced Mac OS X operating system, which had protected memory architecture and pre-emptive multi-tasking, among other benefits. To stay competitive, Microsoft launched Windows XP.

2003: The first 64-bit processor, AMD's Athlon 64, was brought into the consumer market.

2004: Mozilla released Firefox 1.0 and in the same year Facebook, a social networking site, was launched.

2005: YouTube, a video sharing service, was launched. In the same year, Google acquired Android, a Linux-based mobile phone operating system.

2006: Apple introduced MacBook Pro, its first Intel- based, dual-core mobile computer.

2007: Apple released iPhone, which brought many computer functions in the smartphone.

2009: Microsoft launched Windows 7 in which users could pin applications to the taskbar.

2010: Apple launched iPad, which revised the tablet computer segment.

2011: Google introduced Chromebook, a laptop that runs on the Google Chrome operating system.

2015: Apple released the Apple Watch. In the same year, Microsoft launched Windows 10

Let us also understand the evolution of computers through different generations.

First Generation (1942-1955) – Vacuum tubes

Hardware Technology First generation computers were manufactured using thousands of vacuum tubes. A vacuum tube is a device made of fragile glass.

Memory Electromagnetic relay was used as primary memory and punched cards were used to store data and instructions.

Software Technology Programming was done in machine or assembly language.

Used for Scientific applications

Examples ENIAC, EDVAC, EDSAC, UNIVAC LIRM 701

Highlights

- They were the fastest, calculating device of those times
- Computers were too bulky and required a complete room for storage
- Highly unreliable as vacuum tubes emitted a large amount of heat and burnt frequently
- Required air-conditioned rooms for installation
- Costly
- Difficult to use

- Required constant maintenance because vacuum tubes used filaments that had limited life time. Therefore, these computers were prone to frequent hardware failures.

Figure 1.3 Vacuum tube
Source: Vladyslav Danilin/Shutterstock

Second Generation (1955-1964) - Transistors

Hardware Technology Second generation computers were manufactured using transistors. Transistors were reliable, powerful, cheaper, smaller, and cooler than vacuum tubes.

Memory Magnetic core memory was used as primary memory; magnetic tapes and magnetic disks were used to store data and instructions. These computers had faster and larger memory than the first generation computers.

Software Technology Programming was done in high level programming languages. Batch operating system was used.

Used for Scientific and Commercial Applications

Examples Honeywell 400, IBM 7030, CDC 1604, UNIVAC LARC

Highlights

- Faster, smaller, cheaper, reliable, and easier to use than the second generation computers.
- They consumed 1/10 th the power consumed by than first generation computers
- Bulky in size and required a complete room for its installation
- Dissipated less heat than first generation computers but still required air-conditioned rooms
- Costly
- Difficult to use

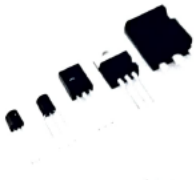


Figure 1.4 Transistors

***Third Generation (1964-1975)* – ICs with SSI, MSI**

Hardware Technology Third generation computers were manufactured using integrated chips (ICs). ICs consist of several components such as transistors, capacitors, and resistors on a single chip to avoid wired interconnections between components. These computers used SSI and MSI technology. Minicomputers came into existence.

Initially, it contained 10-20 components. This technology was called Small Scale Integration (SSI). Later, it was enhanced to contain about 100 components. This was called MSI (Medium Scale integration.

Memory Larger magnetic core memory was used as primary memory: larger capacity magnetic tapes and magnetic disks were used to store data and instructions

Software Technology Programming was done in high level programming languages such as FORTRAN, COBOL Pascal, and BASIC. Time sharing operating system was used. Software was separated from the hardware. This allowed users to invest only in the software they need.

Used for Scientific, commercial, and interactive online applications

Examples IBM 360/370, PDP-8, PADP-11, CDC6600

Highlights

- Faster, smaller, cheaper, reliable, the second generation computers
- They consumed less power than second generation computers
- Bulky in size and required a complete room for its installation
- Dissipated less heat than first generation computers but still required air-conditioned rooms
- Costly
- Easier to use and upgrade

Figure 1.5 Integrated circuits

<u>Fourth Generation (1975-1989) – </u>ICs with LSI, VLSI

Hardware Technology Fourth generation computers were manufactured using ICs with LSI (Large Scale Integrated) and later with VLSI technology (Very Large Scale Integration). Microcomputers came into existence.

Use of personal computers became widespread. High speed computer networks in the form of LAN, WAN, and MANS started growing. Besides mainframes, supercomputers were also used

LSI circuits contained 30,000 components on a single chip and VLSI technology had about one million electronic components on a single chip.

Memory Semiconductor memory was used as primary memory, large capacity magnetic disks were used as built- in secondary memory. Magnetic tapes and floppy disks were used as portable storage devices.

Software Technology Programming done in high level programming language such as C and C++. Graphical User Interface (GUI) based operating system (e.g. Windows) was introduced. It had icons and

menus among other features to allow computers to be used as a general purpose machine by all users. UNIX was also introduced as an open source operating system. Apple Mac OS and MS DOS were also released during this period. All these operating systems had multi-processing and multi- programming capabilities.

Used for Scientific, commercial, interactive online, and network applications

Examples IBM PC, Apple II, TRS-80, VAX 9000, CRAY- 1, CRAY-2, CRAY-X/MP

Highlights Faster, smaller, cheaper, powerful, reliable and easier to use than the previous generation computers.

Figure 1.6 VLSI chip

Fifth Generation (1989-Present) – ICs with ULSI

Hardware Technology Fifth generation computers are manufactured using ICs with ULSI (Ultra Large Scale Integrated) technology. The use of Internet became widespread and very powerful mainframes, desktops, portable laptops, and smartphones are being used commonly Supercomputers use parallel processing techniques.

ULSI circuits contain about 10 million electronic components on a single chip.

Memory Semiconductor memory is used as primary memory; large capacity magnetic disks are used as built-in secondary memory, Magnetic tapes and floppy disks were used as portable storage devices, which have now been replaced by optical disks and USB flash drives.

Software Technology Programming is done in high-level programming languages such as Java, Python, and C#. Graphical User Interface (GUI)-based operating systems such as Windows, Unix, Linux, Ubuntu, and Apple Mac are being used. These operating systems are more powerful and user friendly than the ones available in the previous generations.

Used for Scientific, commercial, interactive online, multimedia (graphics, audio, video), and network applications

Examples IBM notebooks. Pentium PCs, SUM workstations, IBM SP/2, Param supercomputer

Highlights

- Faster, smaller, cheaper, powerful, reliable, and easier to use than the previous generation computers.
- Speed of microprocessors and the size of memory are growing rapidly

- High-end features available on mainframe computers in the fourth generation are now available on the microprocessors.
- They consume less power than computers of prior generations.
- Air-conditioned rooms required for mainframes and supercomputers but not for microprocessors.

Figure 1.7 ULSI chip

1.5 CLASSIFICATION OF COMPUTERS

Computers can be broadly classified into four categories based on their speed, amount of data that they can process and price. These categories are as follows:

- Supercomputers
- Mainframe computers
- Minicomputers
- Microcomputers

1.5.1 Supercomputers

Among the four categories, the supercomputer is the fastest, most powerful, and most expensive computer. Supercomputers were first developed in the 1980s to process large amounts of data and to solve complex scientific problems. Supercomputers use parallel processing technology and can perform more than one trillion calculations in a second.

A single supercomputer can support thousands of users at the same time. Such computers are mainly used for weather forecasting, nuclear energy research, aircraft design, automotive design, online banking, controlling industrial units etc.

1.5.2 Mainframe Computers

Mainframe computers are large-scale computers (but smaller than supercomputers). These are very expensive and need a very large clean room with air conditioning, thereby making them very costly to deploy. As with supercomputers, mainframes can also support multiple processors. For example, the IBM S/390 mainframe can support 50,000 users at the same time. Users can access mainframes by either using terminals or via PCs. The two types of terminals that can be used with mainframe systems are as follows:

Dumb Terminals

Dumb terminals consist of only a monitor and a keyboard (or mouse). They do not have their own CPU and memory and use the mainframe system's CPU and storage devices.

Intelligent Terminals

In contrast to dumb terminals, intelligent terminals have their own processor and thus can perform some processing operations. However, just like the dumb terminals, they do not have their own storage space. Usually, PCs are used as intelligent terminals to facilitate data access and other services from the mainframe system.

1.5.3 Minicomputers

Minicomputers are smaller, cheaper, and slower than mainframes. They are called minicomputers because they were the smallest computer of their times. Also known as midrange computers, the capabilities of minicomputers fall between mainframe and personal computers.

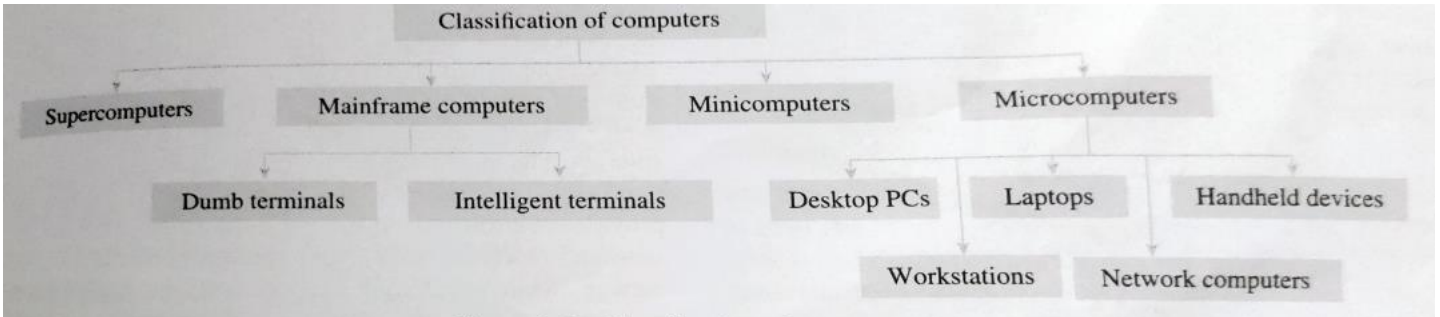


Figure 1.8 Classification of computers

Minicomputers are widely used in business, education, hospital government organizations, etc. While some minicomputers can be used only by a single user, others are specifically designed to handle multiple users simultaneously.

1.5.4 Microcomputers

Microcomputers, commonly known as PCs, are very small and cheap. The first microcomputer was designed by IBM in 1981 and was named IBM-PC, Later on, many computer hardware companies copied this design and termed their microcomputers as PC-compatible, which refers to any PC that is based on the original IBM PC design.

Another type of popular PC is designed by Apple. PCs designed by IBM and other PC-compatible computers have a different architecture from that of Apple computers.

PCs can be classified into the following categories:

Desktop PCs

A desktop PC is the most popular model of PCs. The system unit of the desktop PC can be placed flat on a desk or table. It is widely used in homes and offices.

Laptops

Laptops are small microcomputers that can easily fit inside a briefcase. They are very handy and can easily be carried from one place to another.

Figure 1.9 Laptop
Source: You can more/Shutterstock

Workstations

Workstations are single-user computers that have the same features as PCs, but their processing speed matches that of a minicomputer or mainframe computer. Workstation computers have advanced processors, more RAM and storage capacity than PCs. Therefore, they are more expensive and powerful than a normal desktop computer.

Network Computers

Network computers have less processing power, memory, and storage than a desktop computer. These are specially designed to be used as terminals in a networked environment. For example, some network computers are specifically designed to access data stored on a network (including the Internet and intranet), Some network computers do not have any storage space

Handheld Computers

The mid-1990s witnessed a range of small personal computing devices that are commonly known as handheld computers, or mobile computers. These computers are called handheld computers because they can fit in one hand, while users can use the other hand to operate them.

Handheld computers are very small in size, and hence they have small-sized screens and keyboards. These computers are preferred by business travelers and mobile employees whose jobs require them to move from place to place.

Some examples of handheld computers are as follows:

Smartphones

Tablet PCs

Smartphones These days, cellular phones are web-enabled telephones. Such phones are also known as smartphones because, in addition to basic phone capabilities, they also facilitate the users to access the Internet and send e-mails, edit Word documents, generate an Excel sheet, and create a presentation, and lots more.

Smartphones run an advanced mobile operating system that enables it to run various applications. The four major mobile operating systems are iOS, Android, BlackBerry OS, and Windows Mobile Smartphones also have a CPU, more storage space, more memory, and a larger screen than a regular cell phone.

In a nutshell, smartphone refers to a multi-functional mobile phone handset that packs in varied

functionalities from a camera to a web browser to a high-density display.

Tablet PCs A tablet PC is a computing device that is smaller than a laptop, but bigger than a smartphone. Features such as user-friendly interface, portability, and touch screen have made them very popular in the last few years. These days, a wide range of high-performance tablets are available in the market.

While all of them look similar from outside, they may differ in features such as operating system, speed of data connectivity, camera specifications, size of the screen. Processing power, battery life, and storage capability.

Some operating systems that are used in tablets are Android Jellybean (an open-source operating system built by Google), Windows 8, and iOS (developed by Apple). Each operating system has its own advantages and disadvantages and a proprietary app store, from which users can download applications, extending the tablet's functionality. These apps range from games to specialized word processors and even instruments.

While users can easily type directly on the surface of a tablet, some users prefer a wireless or bluetooth-connected keyboard. These days, tablets also offer an optional docking station with keyboards that transforms the tablet into a full-featured netbook.

Uses The following are the uses of Tablet PCs:

- View presentations
- Videoconferencing
- Reading e-books, e-newspaper
- Watching movies
- Playing games
- Sharing pictures, video, songs, documents, etc.
- Browsing the Internet
- Keeping in touch with friends and family on popular social networks, sending emails
- Business people use them to perform tasks such as editing a document, exchanging documents, taking notes, and giving presentations.
- Tablets are best used in crowded places such as airports and coffee shops, where size and portability become more important.

Note - Tablets may replace laptops if users don't have to perform heavy processing tasks and do not require a CD or DVD player.

Figure 1.10 Tablet

1.6 APPLICATIONS OF COMPUTERS

When the first computers were developed, they were used only in the fields of mathematics and science. Let us discuss how computers are being effectively utilized to perform important tasks.

Word processing Word processing software enables users to read and write documents. Users can also add images, tables, and graphs for illustrating a concept. The software automatically corrects spelling mistakes and includes copy paste features (which is very useful where the same text has to be repeated several times).

Internet The Internet is a network of networks that connects computers all over the world. It gives the user access to an enormous amount of information, much more than available in any library. Using e-mail, the user can communicate in seconds with a person who is located thousands of miles away. Chat software enables users chat with another person in real-time (irrespective of the physical location of that person). Video conferencing tools are becoming popular for conducting meetings with people who are unable to be present at a particular place.

Digital video or audio composition Computers make audio or video composition and editing very simple. This has drastically reduced the cost of equipment to compose music or make a film. Graphics engineers use computers for developing short or full-length films and creating 3-D models and special effects in science fiction and action movies.

Desktop publishing Desktop publishing software enables us to create page layouts for entire books.

After discussing how computers are used in today's scenario, let us now have a look at the **different areas where computers are being widely utilized.**

e-Business e-Business or electronic business is the process of conducting business via the Internet. This may include buying and selling of goods and services using computers and the Internet. Use of email and videoconferencing technology has revolutionized the way business is being conducted these days. Techniques in which e-commerce helps users to conduct business transactions:

Business-to-consumer or B2C In this form of electronic commerce, business companies deploy their websites on the Internet to sell their products and services to the customers. On their websites, they provide features such as catalogues, interactive order processing system, secure

electronic payment system, and online customer support,

Business-to-business or B2B This type of electronic commerce involves business transactions performed between business partners (customers are not involved). For example, companies use computers and networks (in the form of extranets) to order raw materials from their suppliers. Companies can also use extranets to supply their products to their dealers.

Consumer-to-consumer or C2C This type of electronic commerce enables customers to carry business transactions among themselves. For example, on auction websites a customer sells his/her product which is purchased by another customer.

Electronic banking Electronic banking, also known as cyberbanking or online banking, supports various banking activities conducted from home, a business, or on the road instead of a physical bank location.

Bioinformatics

Bioinformatics is the application of computer technology to manage large amount of biological information Computers are used to collect, store, analyze, and integrate biological and genetic information to facilitate gene-based drug discovery and development.

Health care

Last few years have seen a massive growth of computers and smartphone users. Like in our daily lives, computers have also become a necessary device in the health care industry. The following are areas in which computers are extensively used in the health care industry.

Storing records Earlier, patient records were kept on paper, with separate records dealing with different medical issues from separate healthcare organizations. With time, the number of prescriptions, medical reports, etc., grow in volume making it difficult to maintain and analyze.

Surgical procedures Computers are used for certain surgical procedures. They enable the surgeon to use computer to control and move surgical instruments in the patient's body for a variety of surgical procedures.

Better diagnosis and treatment Computers help physicians make better diagnoses and recommend treatments. Moreover, computers can be used to compare expected results with actual results in order to help physicians make better decisions.

Geographic Information System and Remote Sensing

A geographic information system (GIS) is a computer-based tool for mapping and analysing earth's features.

Remote sensing is a sub-field of geography, which can be applied in the following areas to collect data of dangerous or inaccessible areas for the following:

- Monitoring deforestation in areas like the Amazon Basin

- Studying features of glaciers in Arctic and Antarctic regions
- Analysing the depth of coastal and ocean areas
- Studying land usage in agriculture
- Examining the health of indigenous plants and crops
- Determining the prospect for minerals
- Locating and measuring intensity of earthquakes (after they had occurred) by comparing the relative intensity and precise timings of seismograms collected from different locations

Meteorology

Meteorology is the study of the atmosphere. This branch of science observes variables of Earth's atmosphere such as temperature, air pressure, water vapour, and the gradients and interactions of each variable, and how they change over time.

Weather forecasting It includes application of science and technology to predict the state of the atmosphere (temperature, precipitation, etc.) for a future time and a given location. Weather forecasting is done by collecting quantitative data about the current state of the atmosphere and analysing the atmospheric processes to project how the atmosphere will evolve.

Aviation meterology

It studies the impact of weather on air traffic management.

Agricultural meterology

It deals with the study of effects of weather and climate on plant distribution, crop yield, water-use efficiency, plant and animal development.

Nuclear meterology

It studies the distribution of radioactive aerosols and gases in the atmosphere.

Maritime meterology

It is the study of air and wave forecasts for ships operating at sea.

Multimedia and Animation

It combines still images, moving images, text, and sound in meaningful ways is one of most powerful aspects of computer technology. We all have seen cartoon movies, which are nothing but an example of computer animation.Displaying a number of images within a fraction of a second gives an animation effect. For example, displaying at least 30 images in a second gives an effect of a moving image.

Edutainment is the combination of education with entertainment

Legal System

Computers are used by lawyers to shorten the time required to conduct legal precedent and case research. Lawyers use computers to look through millions of individual cases and find whether similar or parallel

cases have been approved, denied, criticized, or overruled in the past. This enables the lawyers to formulate strategies based on past case decisions. Moreover, computers are also used to keep track of appointments and prepare legal documents and briefs in time for filling cases

Retail Business

Computers are used in retail shops to enter orders, calculate costs, and print receipts. They are also used to keep an inventory of the products available and their complete description.

Sports

In sports, computers are used to compile statistics, identify weak players and strong players by analyzing statistics, sell tickets, create training programs and diets for athletes, and suggest game plan strategies based on the competitor's past performance. Computers are also used to generate most of the graphic art displays flashed on scoreboards

Travel and Tourism

Computers are used to prepare tickets, monitor the train's or airplane's route, and guide the plane to a safe landing. They are also used to research about hotels in an area, reserve rooms, or to rent a car.

Simulation

Supercomputers that can process enormous amount of data are widely used in simulation tests. Simulation of automobile crashes or airplane emergency landings is done to identify potential weaknesses in designs without risking human lives. Supercomputers also enable engineers to design aircraft models and simulate the effects that winds and other environmental forces have on those designs.

Astronauts are trained using computer-simulated problems that could be encountered during launch in space, or upon return to earth.

Astronomy

Spacecrafts are usually monitored using computers that not only keep a continuous record of the voyage and of the speed, direction, fuel, and temperature, but also suggest corrective action if the vehicle makes a mistake.

The remote stations on the earth compare all these quantities with the desired values, and in case these values need to be modified to enhance the performance of the spacecraft, signals are immediately sent that set in motion the mechanics to rectify the situation. With the help of computers, all this is done within a fraction of a second.

Education

A computer is a powerful teaching aid and can act as another teacher in the classroom. Teachers use computers to develop instructional material. Teachers may use pictures, graphs, and graphical presentations to easily illustrate an otherwise difficult concept. Moreover, teachers at all levels can use computers to administer assignments and keep track of grades. Students can also give exams online and get instant

results.

Industry and Engineering

Computers are found in all kinds of industries, such as thermal power plants, oil refineries, and chemical industries, for process control, computer-aided designing (CAD), and computer-aided manufacturing (CAM).

Computerized process control (with or without human intervention) is used to enhance efficiency in applications such as production of various chemical products, oil refining, paper manufacture, and rolling and cutting steel to customer requirements.

In CAD, computers and graphics-oriented software are integrated for automating the design and drafting process.

Robotics

Robots are computer-controlled machines mainly used in the manufacturing process in extreme conditions where humans cannot work. For example, in high temperature, high Pressure conditions or in processes that demand very high levels of accuracy.

Decision Support Systems

Computers help managers to analyse their organization's data to understand the present scenario of their business, view the trends in the market, and predict the future of their products.

Expert Systems

Expert systems are used to automate the decision making process in a specific area, such an analysing the credit histories for loan approval and diagnosing a patient's condition for prescribing an appropriate treatment. Expert systems analyse the available data in depth to recommend a course of action. A medical expert system might provide the most likely diagnosis of patient's condition.

1.7 BASIC ORGANIZATION OF A COMPUTER

A computer is an electronic device that performs five major operations:

- Accepting data or instructions (input)
- Storing data
- Processing data
- Displaying results (output)
- Controlling and coordinating all operations inside a computer

Input This is the process of entering data and instructions (also known as programs) into the computer system. The data and instructions can be entered by using different input devices such as keyboard, mouse, scanner, and trackball. Note that computers understand binary language, which consists of only two symbols (0 and 1), so it is the responsibility of the input

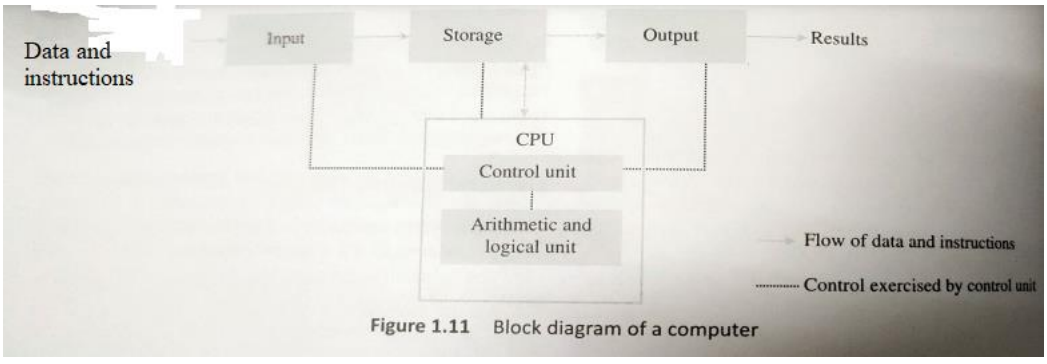


Figure 1.11 Block diagram of a computer

devices to convert the input data into binary codes

Storage - Storage is the process of saving data and instructions permanently in the computer so that they can be used for processing. The computer storage space not only stores the data and programs that operate on that data but also stores the intermediate results and the final results of processing.

A computer has two types of storage areas:

Primary storage - Primary storage, also known as the main memory, is the storage area that is directly accessible by the CPU at very high speeds. It is used to store the data and parts of programs, the intermediate results of processing and the recently generated results of jobs that are currently being worked on by the computer. Primary storage space is very expensive and therefore limited in capacity. Another drawback of main memory is that it is volatile in nature; that is, as soon as the computer is switched off, the information stored gets erased. Hence, it cannot be used as a permanent storage of useful data and programs for future use. An example of primary storage is random access memory (RAM).

Secondary storage - Also known as auxiliary memory. This memory is just the opposite of primary memory. It overcomes all the drawbacks of the primary storage area. It is cheaper, non-volatile, and used to permanently store data and programs of those jobs that are not being currently executed by the CPU. Secondary memory supplements the limited storage capacity of the primary memory. An example is the magnetic disk used to store data, such as C and D drives, for future use.

Output - Output is the process of giving the result of data processing to the outside world (external to the computer system). The results are given through output devices such as monitor, and printer. Since the computer accepts data binary form, the result cannot be directly given to the user. The output devices, therefore, convert the results available in binary codes into a human-readable language before displaying it to the user.

Control - The control unit (CU) is the central nervous system of the entire computer system. It manages and controls all the components of the computer system. It is the CU that decides the manner in which instructions will be executed and operations performed. It takes care of the step-by-step processing of all operations that are performed in the computer.

The CPU is a combination of the arithmetic logic unit (ALU) and the CU. The CPU is better known as the brain of the computer system because the entire processing of data is done in the ALU, and the CU activates and monitors the operations of other units (such as input, output, and storage) of the computer system.

ALU, CU, and CPU are the key functional units of a computer system.

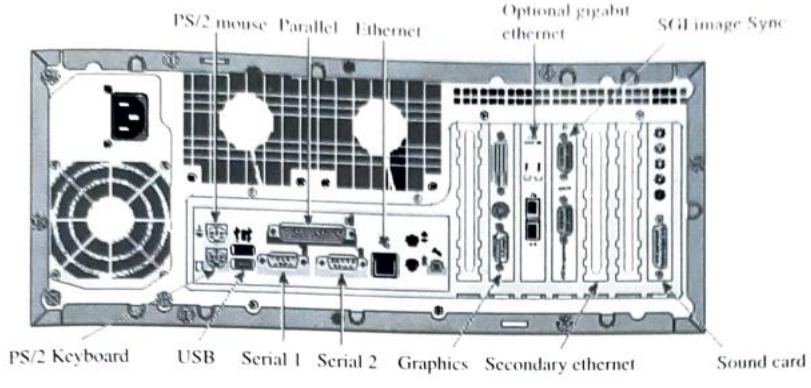
1.8 INSIDE THE COMPUTER

The following are some of the major parts of the computer.

CPU The CPU is the brain of the computer. It performs all calculations and controls the devices connected to the computer system. The faster the CPU, the quicker programs can process the instructions.

RAM A fast CPU is of no use if the computer does not have sufficient RAM. RAM is the computer's memory which stores information used by applications that are currently being executed by the CPU. More memory means more applications can be executed at the same time without degrading the system's performance.

Hard disk drive (HDD) The HDD of the computer is the secondary memory of the computer system where information is stored permanently. All types of data, documents, and programs are stored on the hard disk. The larger the hard disk, the more the amount of data that can be stored on the drive. Though the size of the HDD does not affect the speed of execution of the program, it does affect the speed at which the user can access his/her files.



Computer case and its parts

Video card The video card is a board that plugs into the motherboard of the computer and generates images for display. Many video cards these days have their own RAM and processor to enhance the speed of the graphics display. Many computers come with an in-built video chip. In such a computer, a separate video card is used only if the computer has to be used for high-end multimedia work or to play video games.

Sound card - As with video cards, sound cards are expansion boards that are used to enable a computer to manipulate sound. For example, sound cards allow the users to plug in speakers and a microphone. Some sound cards also provide the jacks for hooking our computer up to a common stereo. These days, many

computers come with a built-in sound chip, which makes it unnecessary to buy a separate card unless a higher quality of sound is needed.

Modem A modem (modulator-demodulator) is a device that enables the computer to use a telephone line to communicate and connect to the Internet.

Network card A network card is used to connect the computer either to other computers or to the Internet (in case we are using a fast Internet connection such as cable or DSL).

Fans There are one or more fans inside the computer to keep the air moving and the computer cool.

Cables There are multiple wires inside the computer that are flat, ribbon-like cables. They are used to provide power and communication to the various parts inside the computer.

1.9 MOTHERBOARD

The motherboard, also known as the mainboard or the parent board, is the primary component of a computer. It is used to connect all the components of the computer. The motherboard is a printed circuit that has connectors for expansion cards, memory modules, the processor, etc.

1.9.1 Characteristics of a Motherboard

A motherboard can be classified depending on the following characteristics:

- Form factor
- Chipset
- Type of processor socket used
- Input-Output connectors

Form factor Form factor refers to the motherboard's and electrical geometry, dimensions, arrangement, requirements. The industry has defined a few standards for the form factors, so that they can be used in different brands of cases.

Integrated components some of the motherboard's components are integrated into its printed circuitry. These include the following:

- The **chipset** is a circuit that controls the majority of the computer's resources such as the bus interface with the processor, cache memory, RAM, and expansion cards.
- CMOS clock and battery
- BIOS
- System bus and expansion bus

In addition to these, the latest motherboards also have a number of onboard multimedia and networking devices (which can be disabled), such as integrated network card, integrated graphics card, integrated sound card, and upgraded hard drive controllers.

Chipset The chipset is an electronic circuit that basically coordinates data transfers between the different components of the computer (such as the processor and memory). In order to enhance the computer's

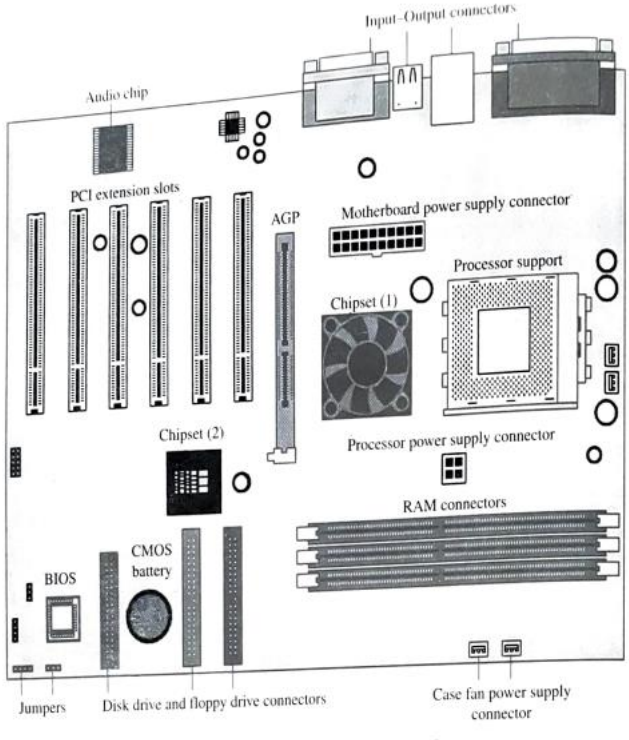
upgradeability, one must choose a motherboard that has the latest chipset integrated in it. Some chipsets may include a graphics or audio chip, which makes it unnecessary to install a separate graphics card or sound card. However, in case we need very high quality of audio and visual capabilities, then we must disable the graphics/audio chip in the BIOS setup and install high-quality expansion cards in the appropriate slots.

CMOS clock and battery The real-time clock (or RTC) is a circuit that is used to synchronize the computer's signals. When the computer is switched off, the power supply stops providing electricity to the motherboard. We must have observed that when we turn on the system, it always displays the correct time. This is because an electronic circuit, called the complementary metal-oxide semiconductor (CMOS) chip, saves some system information, such as the time, date, and other essential system settings.

At times, the system time gets reset automatically, or the clock runs late? This indicates that we need to change the battery.

BIOS The basic input/output system (BIOS) is an interface between the operating system and the motherboard. The BIOS is stored in the read-only memory (ROM), which cannot be rewritten. The BIOS uses data stored in the CMOS chip to know about the system's hardware configuration.

To configure the BIOS, the user can use an interface known as BIOS setup, which can be accessed when the computer is booting. To enter BIOS setup, the user must press the DEL key. Fl and F2 keys can also be used.



Computer's motherboard

Processor socket The processor (also called the micro- processor) is the brain of the computer. The processor is characterized by its speed or frequency, which is the rate at which it executes instructions. For

example, an 800-MHz processor can perform 800 million operations per second.

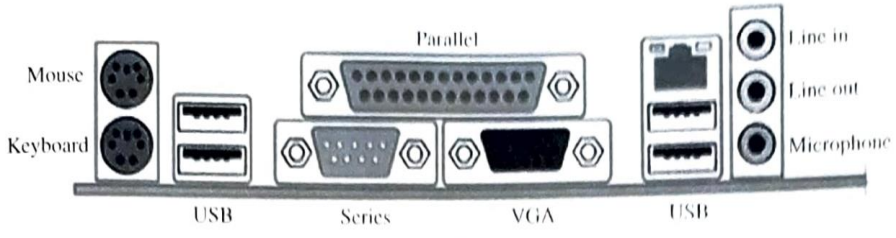
The slot on the motherboard into which the processor is inserted is called the processor socket or slot. Irrespective of whether we use a slot or a socket, we must gently insert the processor, so that none of its pins are bent (it has hundreds of them). Usually, a concept called zero insertion force (ZIF) is used. The ZIF sockets allow the processor to be inserted very gently and easily

RAM connectors RAM is the primary storage area that stores data while the computer is running. However, its contents are erased when the computer is turned off or restarted. While the hard disk can store data permanently, we still need RAM because it is extremely fast when compared to mass storage devices such as hard drives. Therefore, the fast processor accesses data from RAM and not from the hard disk. The data is transferred from the hard disk to the RAM, from where it is used by the processor RAM is available in the form of modules that plug into motherboard connectors.

Expansion slots Expansion slots are compartments into which expansion cards can be inserted. Such cards render new features or enhance the computer's performance. For example, the AGP slot (also known as Accelerated Graphic Port) is a fast port used for graphics cards.

I/O connectors The motherboard has a number of input- output sockets on its rear panel, some of which include:

- A serial port to connect some old peripherals
- A parallel port to connect old printers
- USB ports to connect more recent peripherals such as mouse and pen drive.
- RJ45 connector (also known as LAN or Ethernet port) to connect the computer to a network. It corresponds to a network card integrated into the motherboard.
- Video graphics array (VGA) connector to connect a monitor. This connector interfaces with the built-in graphics card.
- Audio plugs that include the line in, line out, and microphone to connect sound speakers, hi-fi system, or microphone. This connector interfaces with the built-in sound card.



I/O connectors

Chapter 2: Input and Output devices

2.1 INPUT DEVICES An input device is used to feed data and instructions into a computer.

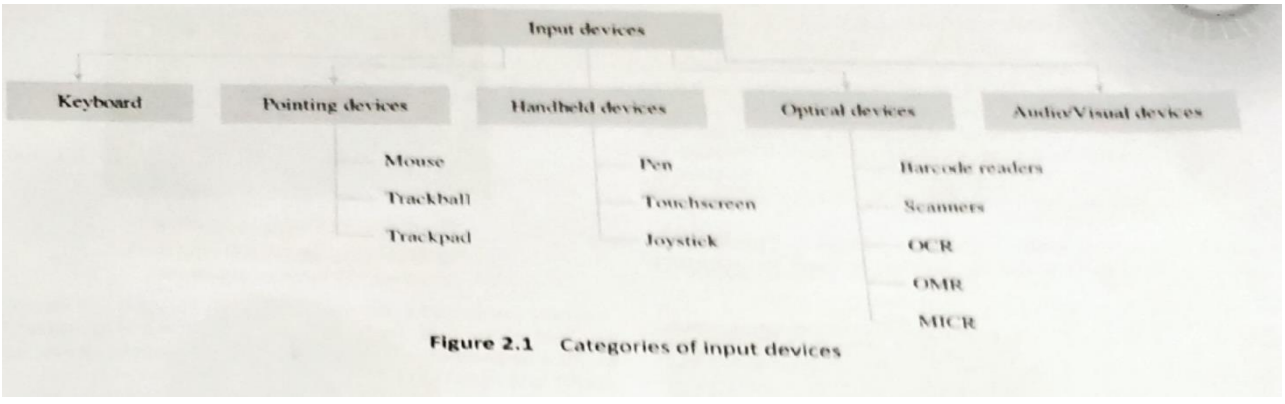


Figure 2.1 Categories of input devices

2.1.1 Keyboard The keyboard is the main input device for computers.

Computer keyboards look very similar to the keyboards of typewriters, with some additional keys.

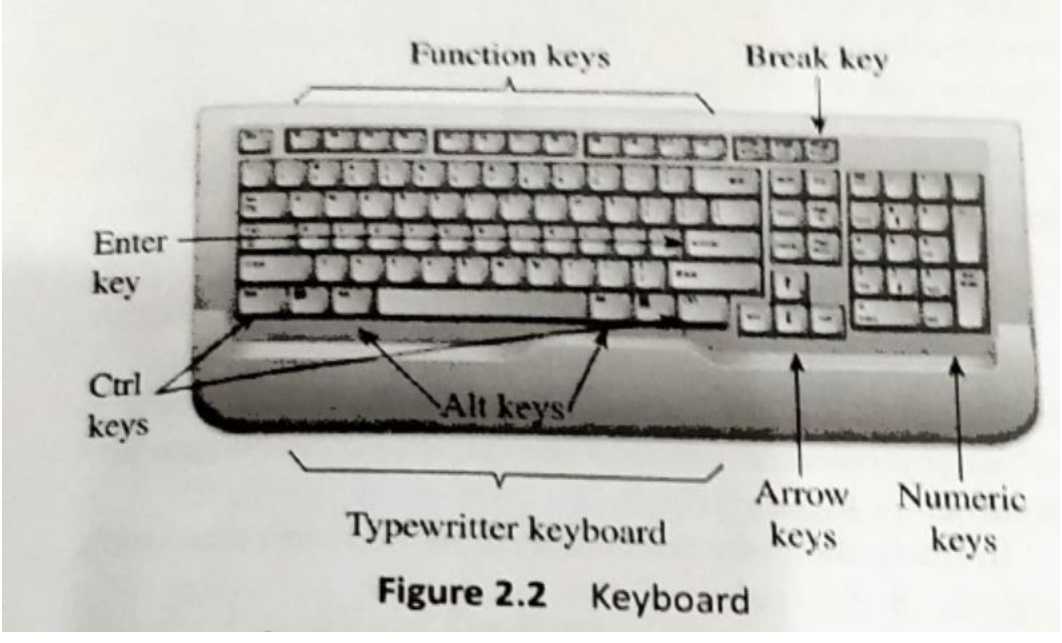


Figure 2.2 Keyboard

Using a keyboard, the user can type a document, use keystroke shortcuts, access menus, play games, and perform numerous other tasks. Most keyboards have between 80 and 110 keys, which include the following:

Typing keys These include the letters of the alphabet. The layout of the keyboard is known as QWERTY for its first six letters. The QWERTY pattern has been a standard right from the time computer keyboards were introduced.

Numeric keys These include a set of keys, arranged in the same configuration found on calculators to speed up data entry of numbers. When the Num Lock key is set to ON, the user can type numbers, dot, or input the symbols/, *, –, and +. When the Num Lock key is set to OFF, the numeric keys can be used to move the cursor on the screen.

Function keys These are used by applications and operating systems to input specific commands. They are

often placed on the top of the keyboard in a single row. Function keys can be programmed so that their functionality varies from one program to another.

Control keys These are used to handle control of the cursor and the screen. Four arrow keys are arranged in an inverted T -type fashion between the typing and the numeric keys, and are used to move the cursor on the screen in small increments. In addition to the arrow keys, there are other cursor keys (or navigational keys), such as:

• Home and End to move the cursor to the beginning and end of the current line, respectively

• Page Up and Page Down to move the cursor up and down by one screen at a time, respectively.

• Insert to enter a character between two existing characters

• Delete to delete a character at the cursor position.

Other common control keys on the keyboard include Control (Ctrl), Alternate (Alt), Escape (Esc), Print Screen, Pause, the Windows or Start key (Microsoft Windows logo), and a shortcut key. The shortcut key is used to access the options available by pressing the right mouse button. The Esc key cancels the selected option, and the Pause key suspends a command/process in progress. Finally, the Print Screen key captures everything on the screen as an image. The image can be pasted into any document.

Note Keys such as Shift, Ctrl, and Alt are called modifier keys because they are used to modify the normal function of a key. For example, Shift + character (lowercase) makes the computer display the character in upper case.

Inside the Keyboard

A keyboard is like a miniature computer that has its own processor and circuitry to carry information to and from that processor. A character map is a lookup table that tells the processor what each keystroke or a combination of keystrokes represents. For example, the character map tells the processor that pressing the key ‘c’ corresponds to a small letter ‘c’ but the keys ‘Shift’ and ‘c’ pressed together is a ‘C’.

Advantage The keyboard is easy to use and inexpensive.

Disadvantages

• The keyboard cannot be used to draw figures.

• The process of moving the cursor to another position is very slow. Mouse and other pointing devices are more apt for this purpose.

2.1.2 Pointing Devices

A pointing input device enables the users to easily control the movement of the pointer to select items on a display screen, to select commands from commands menu, to draw graphics, etc. Some examples of pointing devices include mouse, trackball, light pen, joystick, and touchpad.

Mouse The mouse is an input device that was invented by Douglas Engelbart in 1963. It is the key input device used in a graphical user interface (GUI). It can be used to handle the pointer easily on the screen to

perform various functions such as opening a program or file. With the mouse, the users no longer need to memorize commands, which was earlier a necessity when working with text-based command line environments like MS-DOS.

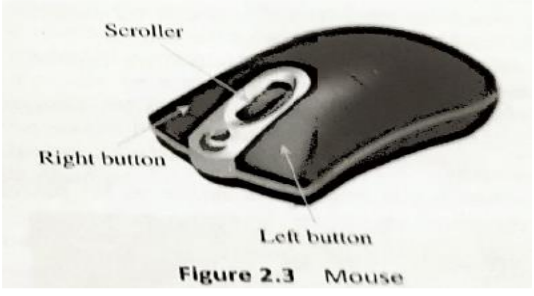


Figure 2.3 Mouse

The mouse has two buttons and a scroll wheel. It can be held in the hand and easily moved, without lifting, along a hard flat surface to move the cursor to the desired location—up, down, left, or right. Once the mouse is placed at the appropriate position, the user may perform the following operations:

Point Placing the pointer over the word or the object on the screen by moving the mouse on the desk is termed as pointing.

Click Pressing either the left or the right button of the mouse is known as clicking. Clicking a mouse button initiates some action; for example, when we click the right button by pointing the mouse on a word, a menu pops up on the screen. When we move the pointer over the icon of an application, say Internet Explorer, and double-click on it, then it opens that application.

Drag Dragging means pointing to a desired location while pressing the left button.

Scroll The scroll wheel, which is placed in between the left and right buttons of the mouse, is used to vertically scroll through long documents.

Some of the popular mouse types are as follows:

Mechanical mouse This type of mouse has a rubber or metal ball at its bottom and an electronic circuit containing sensors. When the mouse is moved over a flat surface, the sensors detect the direction of movement of the ball. The electronic circuit translates the movement into signals and feeds it as input to the computer.

Optical mouse The optical mouse is more advanced than the mechanical mouse. It contains a ball inside. The movement of the mouse is detected using laser technology, by using optical sensors.

Cordless mouse A cordless or wireless mouse is not connected to the computer. The movement of the mouse is detected using radio waves or infrared light waves.

Advantages

• The mouse is easy to use and can be used to quickly place the cursor anywhere on the screen.

• It also helps to quickly and easily draw figures.

• It is inexpensive.

• Its point-and-click capabilities make it unnecessary to remember and type in commands.

Disadvantages

• The mouse needs extra desk space to be placed and moved easily.

• The ball in the mechanical mouse must be cleaned to remove dust from it.

Trackball

A trackball is a pointing device that is used to control the position of the cursor on the screen. It is usually used in notebook computers, where it is placed on the keyboard. The trackball is nothing but an upside-down mouse where the ball rotates in place within a socket. The user rolls the ball to position the cursor at an appropriate position on the screen and then clicks one of the buttons (identical to mouse buttons) near the trackball, either to select objects or to position the cursor for text entry.

To move the pointer, the ball is rotated with the thumb, fingers, or the palm of the hand. The advantage of a trackball over a mouse is that the former is stationary, and so it does not require much space to use. Moreover, individual trackballs can be placed on any type of surface, including the user's lap. These advantages make trackballs very popular pointing devices for portable computers and mobile phones. The working of a trackball is identical to that of mouse.

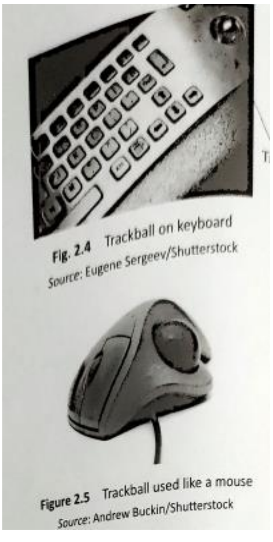


Fig. 2.4 Trackball on keyboard
Source: Eugene Sergeev/Shutterstock

Figure 2.5 Trackball used like a mouse
Source: Andrew Buckin/Shutterstock

Advantages

• The trackball provides better resolution.

• It occupies less space.

• It is easier to use as compared to a mouse as its use involves less hand and arm movements.

Disadvantage The trackball chamber is often covered with dust, so it must be cleaned regularly.

Touchpad A touchpad (or trackpad) is a small, flat, rectangular stationary pointing device with a sensitive surface of 1.5–2 square inches. The user has to slide his or her fingertips across the surface of the pad to point to a specific object on the screen.

Figure 2.6 Touchpad

Advantages

• Touchpads occupy less space.

• They are easier to use as compared to a mouse as their use involves less hand and arm movements.

• A touchpad is in-built in the keyboard, and hence negates the need to carry an extra device.

2.1.3 Handheld Devices

A handheld device is a pocket-sized computing device with a display screen and touch input and/or a miniature keyboard. Some common examples of handheld devices include smartphones, PDAs, handheld game consoles, and portable media players (e.g., iPods).

Joystick A joystick is a cursor control device widely used in computer games and computer aided design (CAD)/ computer - aided manufacturing (CAM) applications. It consists of a handheld lever that pivots on one end and transmits its coordinates to a computer. A joystick has one or more push buttons, called switches, whose position can also be read by the computer. The lever of a joystick moves in all directions to control the movement of the pointer on the computer screen. A joystick is similar to a mouse, but with the mouse, the cursor stops moving as soon as you stop moving the mouse. However, in case of a joystick, the pointer continues moving in the direction to which the joystick is pointing. To stop the pointer, the user must return the joystick to its upright position.

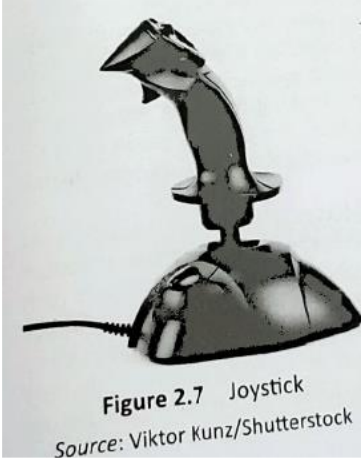


Figure 2.7 Joystick
Source: Viktor Kunz/Shutterstock

Stylus A stylus is a pen-shaped input device used to enter information or write on the touchscreen of a handheld device. It is a small stick that can also be used to draw lines on a surface as input into a device, choose an option from a menu, move the cursor to another location on the screen, take notes, and create short messages. The stylus usually slides into a slot built into the device for that purpose.

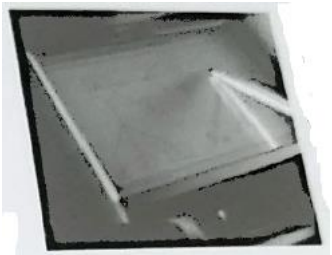


Figure 2.8 Stylus
Source: Photodisc/OUP Picture Bank

Touchscreen A touchscreen is a display screen that can identify the occurrence and position of a touch inside the display region. The user can touch the screen either by using a finger or a stylus. The touchscreen facilitates the users to interact with what is displayed on the screen in a straightforward manner, rather than in an indirect way by using a mouse or a touchpad. Such touchscreen displays are available on computers, laptops, PDAs, and mobile phones. These days, touchscreen monitors are widely used in different applications including point-of-sale (POS) cash registers, PDAs, automated teller machines (ATMs), car navigation screens, mobile phones, gaming consoles, and any other type of appliance that requires the user to input and receive information instantly.

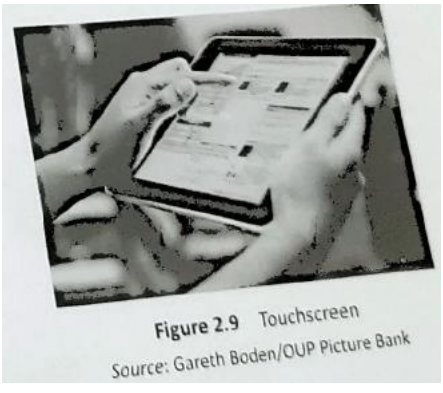


Figure 2.9 Touchscreen
Source: Gareth Boden/OUP Picture Bank

2.1.4 Optical Devices

Optical devices, also known as data-scanning devices, use light as a source of input for detecting or recognizing different objects such as characters, marks, codes, and images.

Barcode Reader A barcode reader (also price scanner or POS scanner) is a handheld input device that is used to capture and read information stored in a barcode. It consists of a scanner, a decoder, and a cable used to connect the reader to a computer. The function of the barcode reader is to capture and translate the barcode into numerals and/or letters. It is connected to a computer for further processing of the captured information. This connection is achieved through a serial port, keyboard port, or an interface device called a wedge.

These days, barcode readers are widely used in following areas:

- Generate bills in supermarkets and retail stores
- Take stock of inventory in retail stores
- Check out books from a library

• Track manufacturing and shipping movement

• Keep track of employee login

• Identify hospital patients

• Tabulate the results of direct mail marketing returns

• Tag honeybees used in research

Advantages

• Barcode readers are inexpensive.

• They are portable.

• They are handy and easy to use.

Disadvantages

• Barcode readers must be handled with care. If they develop a scratch, the user may not be able to read the code.

• They can interpret information using a limited series of thin and wide bars. To interpret other unique identifiers, the bar display area must be widened.

Image Scanner

A scanner is a device that captures images, printed text, and handwriting, from different sources such as photographic prints, posters, and magazines and converts them into digital images for editing and display on computers.

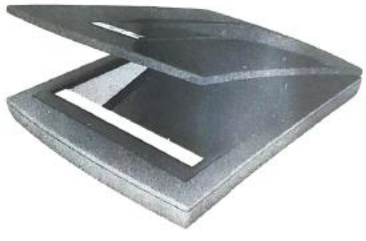


Figure 2.10 Flatbed Image scanner
Source: Mile Atanasov/Shutterstock

Advantages

• Any printed or handwritten document can be scanned and stored in a computer for further processing.

• The scanned and stored document will never deteriorate in quality with time. The document can be displayed and printed whenever required.

• There is no fear of loss of documents. The user can scan important documents and store them permanently in the computer.

Disadvantages

• Scanners are usually costlier than other input devices.

• The documents that are scanned and stored as images have a higher size as compared to other equivalent

text files.

• Text documents are scanned and stored as images. Therefore, they occupy more space and are also un-editable because computers cannot interpret individual characters and numbers in the image.

Optical Character Recognition (OCR) Device

Optical character recognition is the process of converting printed materials into text or word processing files that can be easily edited and stored.

The steps involved in OCR include:

• Scanning the text character by character

• Analysing the scanned image to translate the character images into character codes (e.g., ASCII).

These days, OCR is widely used in the following areas:

• Digitize and preserve documents in libraries

• Process checks and credit card slips

• Sort letters for speeding up mail delivery

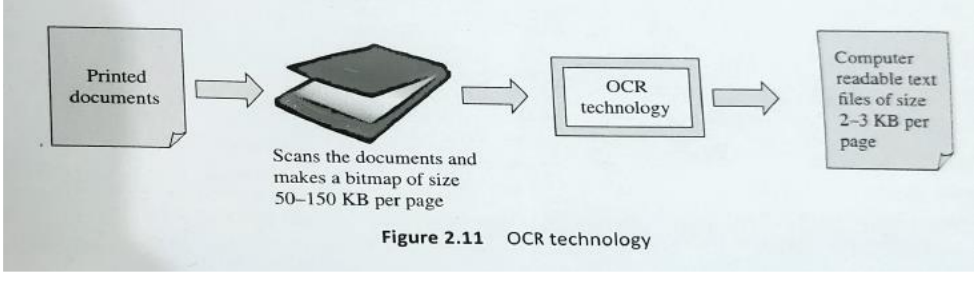


Figure 2.11 OCR technology

Advantages

• Printed documents can be converted into text files.

• Advanced OCR can recognize handwritten text and convert it into computer-readable text files.

Disadvantages

• OCR cannot recognize all types of fonts.

• Documents that are poorly typed or have strikeover cannot be recognized.

• Very old documents when passed through OCR may not convert into an exact copy of the text file. This is because some characters may not have been recognized properly. In such cases, the user has to manually edit the file.

Optical Mark Recognition (OMR) Device

Optical mark recognition is the process of electronically extracting data from marked fields, such as checkboxes and fill-in fields, on printed forms.

The optical mark reader, is fed with an OMR sheet that has pen or pencil marks in pre-defined positions to indicate each selected response (e.g., answers for multiple-choice questions in an entrance examination). The OMR sheet is scanned by the reader to detect the presence of a mark by measuring the reflected light levels. The dark or the marked areas reflect less light than the unmarked ones. The OM reader interprets

this pattern of marks and spaces, and stores the interpreted data in a computer for storage, analysis, and reporting. The error rate for OMR technology is less than 1%. For this reason, OMR is widely used for applications in which large numbers of hand-filled forms have to be quickly processed with great accuracy, such as surveys, reply cards, questionnaires, ballots, or sheets for multiple-choice questions.

Advantage

Optical mark readers work at very high speeds. They can read up to 9000 forms per hour.

Disadvantages

• It is difficult to gather large amounts of information using an OMR.

• Some data may be missing in the scanned document.

• It is a sensitive device that rejects the OMR sheet if it is folded, torn, or crushed.

Magnetic Ink Character Reader

Magnetic ink character reader (MICR) is used to verify the legitimacy of paper documents, especially bank checks. It consists of magnetic ink printed characters that can be recognized by high-speed magnetic recognition devices. The printed characters provide important information (such as cheque number, bank routing number, account number, and, in some cases, the amount on the cheque) for processing to the receiving party.

Audiovisual Input Devices In addition to having a keyboard and a mouse, audio–video devices have become a necessity today.

Audio Devices Audio devices are used to either capture or create sound. They enable computers to accept music, speech, or sound effects for recording and/or editing. Microphones and CD players are examples of two widely used audio input devices. A microphone feeds audio input to the computer. However, the audio input must be converted into digital data before being stored in the computer.

Video Input Devices Video input devices are used to capture video from the outside world into the computer. Here, the term video means moving picture along with sound (as in television). Digital camera and web camera are popular examples of video input devices. A digital camera is a handheld and easily portable device used to capture images or videos. The digital camera digitizes image or video (converts them into 1s and 0s) and stores them on a memory card. The data can then be transferred to the computer using a cable that connects the computer to the digital camera. Once the images or videos are transferred to the computer, they can be easily edited, printed, or transmitted (e.g., through e-mails).

Advantages

• Audio devices can be used by people who are visually impaired.

• Audio input devices are best used in situations where users want to avoid input through keyboard or mouse.

• Video input devices are very useful for applications such as videoconferencing.

• They can be used to record memorable moments in one's life.

• They can also be used for security purposes.

Disadvantages

• Audio input devices are not effective in noisy places.

• With audio input devices, it is difficult to clearly distinguish between two similar sounding words such as 'sea' and 'see'.

• Videos and images captured using video input devices have very big file sizes, and they must be compressed before being stored on the computer.

2.2 OUTPUT DEVICES

Any device that outputs/gives information from a computer can be called an output device. Basically, output devices are electromechanical devices that accept digital data (in the form of 0s and 1s) from the computer and convert them into human-understandable language.

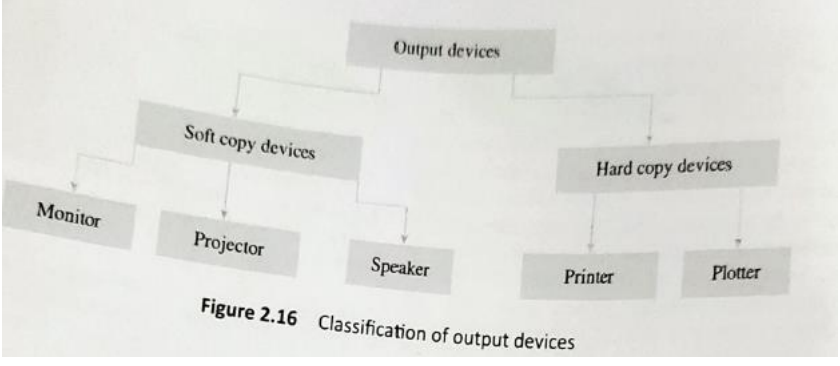


Figure 2.16 Classification of output devices

2.2.1 Soft Copy Devices Soft copy output devices are those that produce an electronic version of an output—for example, a file that is stored on a hard disk, CD, or pen drive—and is displayed on the computer screen (monitor).

Features of a soft copy output include the following:

• The output can be viewed only when the computer is on.

• The user can easily edit soft copy output.

• Soft copy cannot be used by people who do not have a computer.

• Searching for data in a soft copy is easy and fast.

• Electronic distribution of material as soft copy is cheaper. It can be done easily and quickly.

Monitors The monitor is a soft copy output device used to display video and graphics information generated by the computer through the video card. Computer monitors are similar to television screens but they display information at a much higher quality. Monitors come in three variants—cathode ray tube (CRT), liquid crystal display (LCD), and plasma.

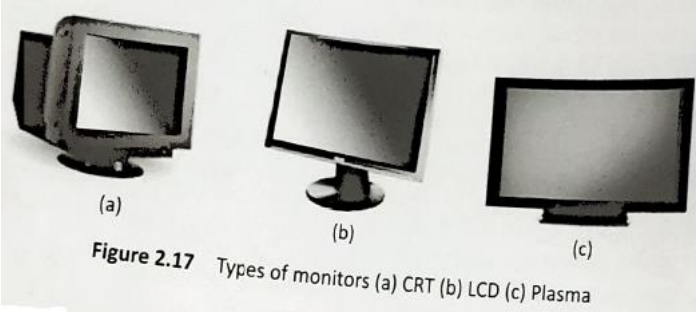


Figure 2.17 Types of monitors (a) CRT (b) LCD (c) Plasma

CRT monitor CRT monitors work by firing charged electrons at a phosphorus film. When electrons hit the phosphor-coated screen, they glow, thereby enabling the user to see the output.

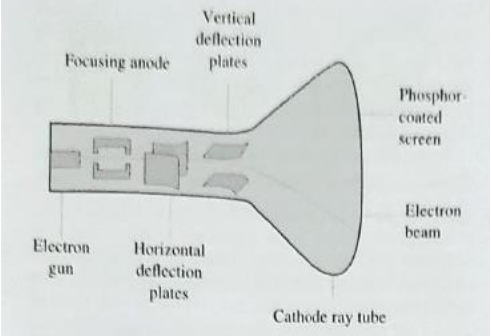


Figure 2.18 Working of a CRT monitor

Colour CRT monitors contain three electron guns (one each for red, blue, and green). Each pixel or dot on the screen has three phosphors (red, blue, and green). When the beam from these guns is focused on the phosphors, they light up. By varying the intensities of the beam, the user can obtain different colours.

Advantages

• CRT monitors provide images of good quality (bright as well as clear).

• CRT monitors are cheapest when compared to LCD and plasma monitors.

• The images are clear even when you try to view it from an angle.

Disadvantages

• CRT monitors occupy a large space on the desk.

• They are bigger in size and weight and therefore difficult to move from one place to another when compared with other types of monitors.

• Power consumption is higher than the other monitors.

LCD Monitor

An LCD monitor is a thin, flat, electronic visual display that uses the light modulating properties of liquid crystals, which do not emit light directly. LCD screens are used in a wide range of applications ranging from computer monitors, televisions, instrument panels, aircraft cockpit displays, signage, etc., to consumer devices such as video players, gaming devices, clocks, watches, calculators, and telephones. Most LCD displays use active matrix technology in which a thin film transistor (TFT) arranges tiny transistors and capacitors in a matrix on the glass of the display. To refer to a particular pixel, the proper

row is turned on, and then a charge is sent through the correct column. Since all the other rows are switched off, only the capacitor at the designated pixel receives a charge.

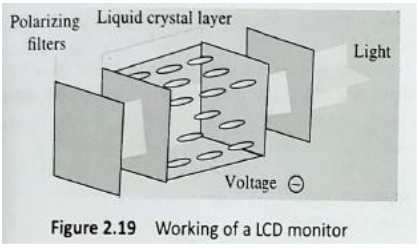


Figure 2.19 Working of a LCD monitor

Advantages

• LCD monitors are very compact and lightweight.

• They consume less power.

• They do not suffer from geometric distortion.

• There is little or no flicker of images (depending on the backlight technology used).

• They are more reliable than CRTs.

• They can be made in almost any size or shape.

• They cause less eye fatigue.

Disadvantages

• They are more expensive than CRTs.

• Images are not very clear when tried to view from an angle.

Plasma monitor Plasma monitors are thin and flat monitors widely used in televisions and computers. The plasma display contains two glass plates that have hundreds of thousands of tiny cells filled with xenon and neon gases.

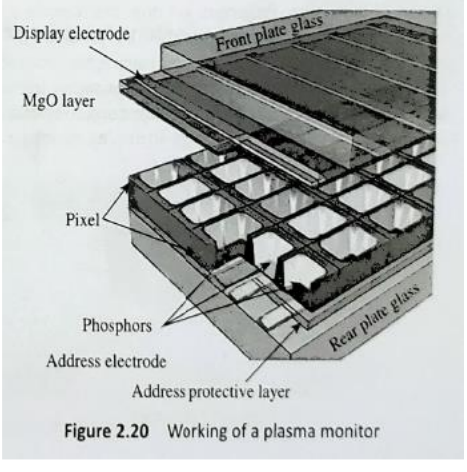


Figure 2.20 Working of a plasma monitor

Advantages

• The technology used in plasma monitors allows producing a very wide screen using extremely thin materials.

• Very bright images are formed which look good from almost every angle.

• These monitors are not heavy and are thus easily portable.

Disadvantages

• These monitors are very expensive.

• They have a high power consumption.

• Since the images are phosphor-based, at times, they may suffer from flicker.

Projectors

A projector is a device that takes an image from a video source and projects it onto a screen or another surface. These days, projectors are used for a wide range of applications, varying from home theater systems for projecting movies and television programs onto a screen much larger than even the biggest available television, to organizations for projecting information and presentations onto screens large enough for rooms filled with many people. Projectors are now available in a variety of different shapes and sizes, and are produced by many different companies.

Projectors can be broadly classified into two categories depending on the technology they use.

LCD projector LCD projectors make use of their own light to display the image on the screen/wall. These projectors are based on LCD technology. To use these projectors, the room must be first darkened, else the image formed will be blurred.

Digital light processing (DLP) projector DLP projectors use a number of mirrors to reflect the light. When using the DLP projector, the room may or may not be darkened because it displays a clear image in both situations.

Speakers Today, all business and home users demand audio capabilities from their computers. For this purpose, speakers were developed in different sizes and shapes, and with different powers and sound quality.

Headphones are small devices that fit in or on the ear, and give about the same quality and power of the sound, as the speakers, only to the listener. Most of today's headphones feature some noise-cancelling technologies, so that the listener may listen to only the sound from the speakers and not anything else from the surrounding environment. Users often use headphones to chat with people over the Internet. With headphones, they are assured that the conversation is heard only by them. However, in addition to the headphones, they are also required to use a separate microphone to talk to the other person. Hence, another device called the headset was developed to allow users to talk and listen at the same time, using the same device. Headsets are widely used in call centers and other telephone-intensive jobs, and for personal use on the computer to facilitate comfortable simultaneous conversation and typing.

2.2.2 Hard Copy Devices Hard copy output devices are those that produce a physical form of output. For example, the content of a file printed on paper is a form of hard copy output. The features of hard copy output include:

• A computer is not needed to see the output.

• Editing and incorporating the edits in the hard copy is difficult.

• Hard copy output can be easily distributed to people who do not have a computer.

• Searching for data in a hard copy is a tiring and difficult job.

• Distribution of hard copy is not only costly but slower as well.

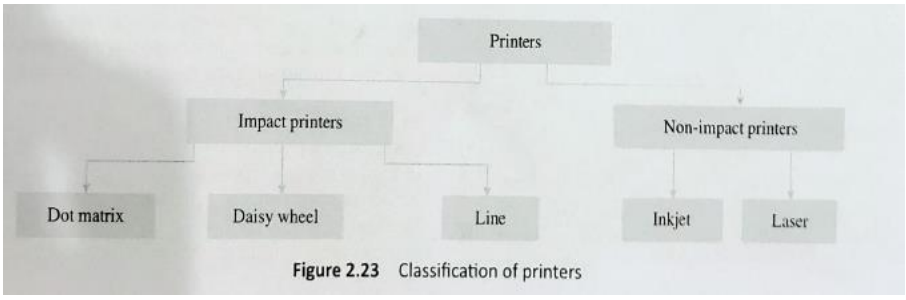


Figure 2.23 Classification of printers

Printers A printer is a device that takes the text and graphics information obtained from a computer and prints it on a paper. Printers are available in the market in various sizes, speeds, sophistication, and costs. Usually, more expensive printers are used for higher-resolution colour printing. The qualities of printers that are of interest to users include:

Colour Colour printouts are needed for presentations, maps, and other pages where colour is part of the information. Colour printers can also be set to print only in monochrome. These printers are more expensive, so if the users do not have a specific need for colour and usually take a lot of printouts, they will find a black-and-white printer cheaper to operate.

Resolution The resolution of a printer means the sharpness of text and images rendered on paper. It is usually expressed in dots per inch (dpi). Even the least expensive printer provides sufficient resolution for most purposes at 600 dpi.

Speed Speed means number of pages that are printed in one minute. The speed of a printer is an important factor for users who have a large number of pages to print. While high-speed printers are quite expensive, the inexpensive printers, on the other hand, can print only about 3–6 sheets per minute. Colour printing is even slower.

Memory Most printers have a small amount of memory (for example, 1 MB), which can be expanded by the user. Having more memory enhances the speed of printing.

Printers can be broadly classified into two groups: impact and non-impact printers.

Impact Printer These printers print characters by striking an inked ribbon against the paper. Examples of impact printers include dot matrix printers, daisy wheel printers, and most types of line printers.

Advantages

• These printers enable the user to produce carbon copies.

• They are cheap.

Disadvantages

• Impact printers are slow.

• They offer poor print quality, especially in the case of graphics.

• They can be extremely noisy.

• They can print only using the standard font.

Non-impact printer Non-impact printers are much quieter than impact printers, as their printing heads do not strike the paper. They offer better print quality, faster printing, and the ability to create prints that contain sophisticated graphics. The main types of non-impact printers are inkjet, laser, and thermal printers.

Advantages

• Non-impact printers produce prints of good quality, and hence render sophisticated graphics.

• They are noiseless.

• They are fast.

• They can print text in different fonts.

Disadvantages

• These printers are expensive.

• The ink cartridges used by them are also costly.

Dot matrix printer

A dot matrix printer prints characters and images of all types as a pattern of dots (hence the name). This printer has a print head (or hammer) that consists of pins representing the character or image. The print head runs back and forth, or in an up-and-down motion on the page and prints by striking an ink-soaked cloth ribbon against the paper, much like the print mechanism of a typewriter.

The speed of dot matrix printers varies in the range of 50–500 cps (characters per second).

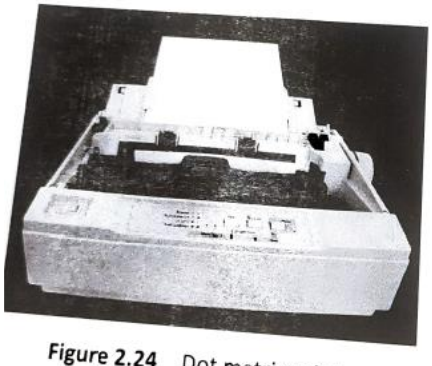


Figure 2.24 Dot matrix printer

Source: burnel1/Shutterstock

Advantages

• The dot matrix printer can produce carbon copies.

• It offers the lowest printing cost per page.

• It is widely used for bulk printing where the quality of the print is not of much importance.

• It is inexpensive.

• When the ink is about to be exhausted, the printout gradually fades rather than suddenly stopping partway through a job.

• It can use continuous paper rather than individual sheets, making them useful for data logging.

Disadvantages

• This type of printer creates a lot of noise when the pins strike the ribbon against the paper.

• It can only print lower resolution graphics, with limited quality.

• It is very slow.

• It has poor print quality.

Daisy wheel printer Daisy wheel printers use an impact printing technology to generate high quality output comparable to typewriters, and are three times faster. However, today, daisy wheel technology is found only in some electronic typewriters. The printhead of a daisy wheel printer is a circular wheel, about 3 inches in diameter with arms or spokes. The shape of the printer wheel resembles the petals of a daisy flower, and hence its name. The characters are embossed at the outer ends of the arms.

The key benefit of using a daisy wheel printer is that the print quality is high, as the exact shape of the character hits the ribbon to leave an impression on the paper.

Line printer A line printer is a high-speed impact printer in which one typed line is printed at a time. The speed of a line printer usually varies from 600 to 1200 lines per minute, or approximately 10–20 pages per minute. Because of their high speed, line printers are widely used in data centers and in industrial environments. Band printer is a commonly used variant of line printers.

Band printer A band printer (loop printer), is an impact printer with a printing mechanism that uses a metal loop or band to produce typed characters.

This type of printer can print 2000 lines per minute, and is, therefore, perfect for high volume printing in businesses, schools, and other organizations. Band printers are normally attached to mainframes and used for industrial printing.

Inkjet printers The printhead of inkjet printers has several tiny nozzles, also called jets. As the paper moves past the printhead, the nozzles spray ink onto it, forming characters and images. If we observe a printout that has just come out from an inkjet printer, we will see that the dots are extremely small (usually between 50 and 60 microns in diameter) and are positioned very precisely, with resolutions of up to 1440 × 720 dpi. To create a coloured image, the dots can have different colours combined together. An inkjet printer can produce from 100 to several hundred pages (depending on the nature of the hard copy), before the ink cartridges must be replaced. There is usually one black ink cartridge and one colour cartridge containing ink in primary pigments (cyan, magenta, and yellow).

Figure 2.25 Inkjet printer

Laser printer A laser printer is a non-impact printer that works at very high speeds and produces high-quality text and graphics. It uses the technology used in photocopier machines.

When a document is sent to the printer, the following steps take place:

• A laser beam 'draws' the document on a drum (which is coated with a photo-conductive material) using electrical charges.

• After the drum is charged, it is rolled in a toner (a dry powder type of ink).

• The toner sticks to the charged image on the drum.

• The toner is transferred onto a piece of paper and fused to the paper with heat and pressure.

• After the document is printed, the electrical charge is removed from the drum and the excess toner is collected.

While colour laser printers are also available in the market, users mostly prefer monochrome printers, because the former is up to ten times more expensive than the latter.

Figure 2.26 Laser printer

Plotters A plotter is a printing device that is usually used to print vector graphics with high print quality. They are widely used to draw maps, in scientific applications, and computer-aided engineering (CAE). Architects use plotters to draw blueprints of the structures they are working on.

Figure 2.27 Plotter

Drum plotter A drum plotter is used to draw graphics on paper that is wrapped around a drum. This type of plotter is usually used with mainframe and minicomputer systems.

Flatbed plotter In a flatbed plotter, the paper is spread on the flat rectangular surface of the plotter, and the pen is moved over it. Flatbed plotters are less expensive, and are used in many small computing systems. The size of the plot is limited only by the size of the plotter's bed. In this type of plotter, the paper does not move; rather, plotting is done by moving an arm that moves a pen over the paper.

Chapter 3: Designing Efficient Programs

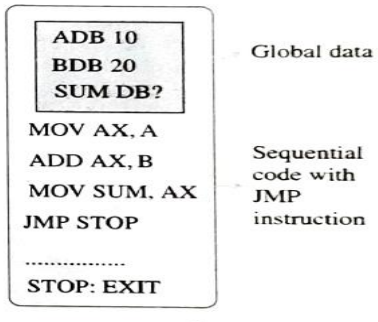
3.1 PROGRAMMING PARADIGMS

A programming paradigm is a fundamental style of programming that defines how the structure and basic elements of a computer program will be built. These paradigms, in sequence of their application, can be classified as follows:

- Monolithic programming - emphasizes on finding a solution
- Procedural programming - lays stress on algorithms
- Structured programming - focuses on modules
- Object-oriented programming - emphasizes on classes and objects
- Logic-oriented programming - focuses on goals usually expressed in predicate calculus
- Rule-oriented programming - makes use of 'if-then- else' rules for computation
- Constraint-oriented programming - utilizes invariant relationships to solve a problem

Monolithic Programming – (emphasizes on finding a solution)

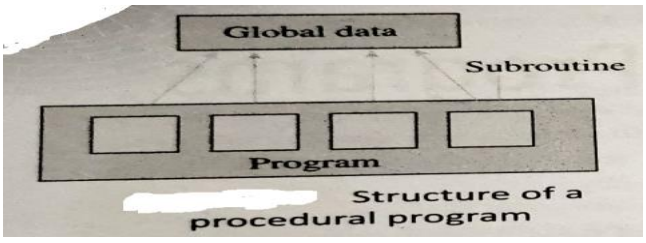
Programs written using monolithic programming languages such as assembly language and BASIC consist of global data and sequential code. The global data can be accessed and modified (knowingly or mistakenly) from any part of the program, thereby posing a serious threat to its integrity. A sequential code is one in which all instructions are executed in the specified sequence. In order to change the sequence of instructions, jump statements or "goto' statements are used.



Structure of a monolithic program

Procedural Programming - lays stress on algorithms

In procedural languages, a program is divided into subroutines that can access global data. To avoid repetition of code, each subroutine performs a well-defined task. With jump, goto, and call instructions. the sequence of execution of instructions can be altered. FORTRAN and COBOL are two popular procedural programming languages.



Advantages

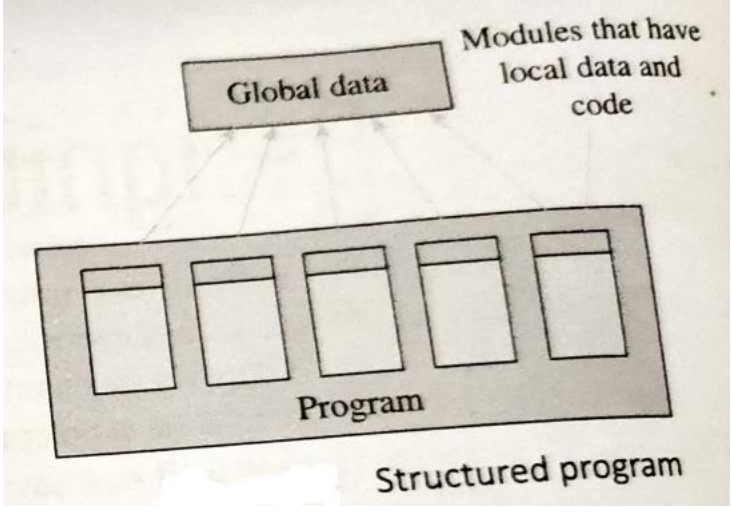
- The only goal is to write correct programs
- Programs are easier to write as compared to monolithic programming

Disadvantages

- No concept of reusability
- Requires more time and effort to write programs
- Programs are difficult to maintain
- Global data is shared and therefore may get altered (mistakenly)

Structured Programming - focuses on modules

Structured programming, also referred to as modular programming. Structured programming employs a top-down approach in which the overall program structure is broken down into separate modules. Modules are coded separately and once a module is written and tested individually, it is then integrated with other modules to form the overall program structure. Almost every modern programming language similar to C, Pascal, supports the concepts of structured program.



Besides sequence, structured programming also supports selection and repetition.

Selection - allows for choosing any one of a number of statements to execute, based on the current status of the program. Ex- if, else, switch.

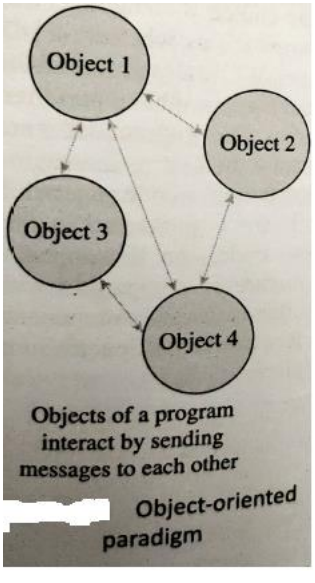
Repetition – A selection statement remains active until the program reaches a point where there is a need for some other action to takes place. Ex – while, do, for

Advantages

- The goal of structured programming is to write correct programs that are easy to understand and change.
- Modules enhance programmers' productivity by allowing them to look at the big picture first and focus on details later.
- With modules, many programmers can work on a single, large program, with each working on a different module.
- A structured program takes less time to be written than other programs. Modules or procedures written for one program can be reused in other programs as well.
- Each module performs a specific task.
- Each module has its own local data.
- A structured program is easy to debug.
- Individual procedures are easy to change as well as understand.
- More emphasis is given on the code and the least importance is given to the data.

  Disadvantages
- Not data centered
- Global data is shared and therefore may get modified
- Main focus is on functions

Object-oriented programming - emphasizes on classes and objects



In this paradigm, all the relevant data and tasks are grouped together in entities known as objects. For example, consider a list of numbers- In the object-oriented paradigm, the list and the associated operations are treated as one entity known as an object. In this approach, the list is considered an object consisting of the list, along with a collection of routines for manipulating the list. In the list object, there may be routines for adding a number to the list, deleting a number from the list, sorting the list, etc.

The striking features of OOP include the following :

- Programs are data centered.
- Programs are divided in terms of objects and not procedures.
- Functions that operate on data are tied together with the data.
- Data is hidden and not accessible by external functions.
- New data and functions can be easily added as and when required.
- Follows a bottom-up approach for problem solving

3.2 EXAMPLE OF A STRUCTURED PROGRAM

We want to create a program to manage the names and addresses of a list of students. For this, we would need to break down the program into the following modules:

- Enter new names and addresses
- Modify existing entries
- Sort entries
- Print the list

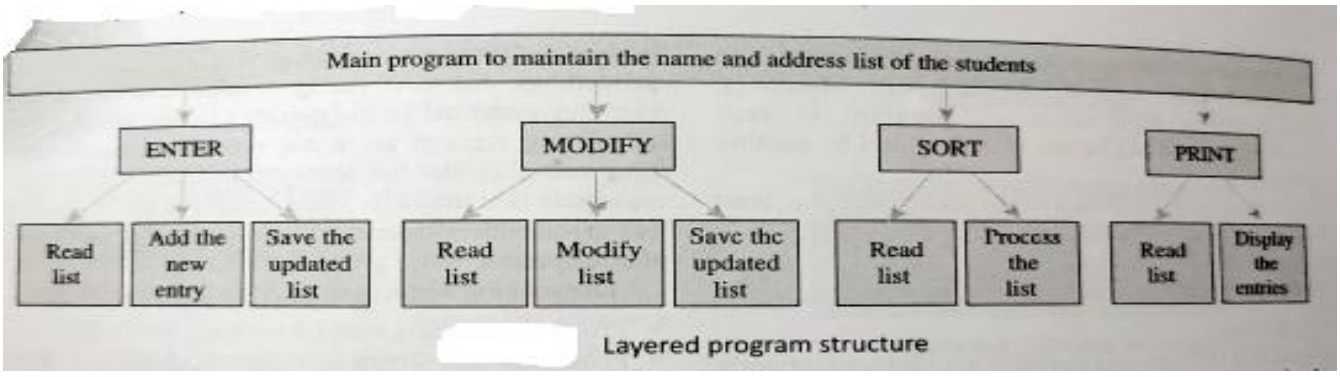
Now, each of these modules can be further broken down into smaller modules. For example, the first module can be subdivided into modules such as follows:

- Prompt the user to enter new data
- Read the existing list from the disk
- Add the name and address to the existing list
- Save the updated list to the disk

Similarly, 'Modify existing entries can be further divided into modules such as follows:

- Read the existing list from disk
- Modify one or more entries
- Save the updated list to disk

The two sub-modules Read the existing list from disk' and 'Save the updated list to disk' are common to both the modules. Hence, once these sub-modules are written, they can be used in both the modules, which require the same tasks to be performed. The structured programming method thus results in a hierarchical or layered program structure.



Layered program structure

3.3 DESIGN AND IMPLEMENTATION OF EFFICIENT PROGRAMS (SDLC)

The entire program or software (collection of programs) development process is divided into a number of phases, where each phase performs a well- defined task. Moreover, the output of one phase provides the input for its subsequent phase.

The phases in the SDLC process can be summarized as follows:

Requirements analysis In this phase, the user's expectations are gathered to know why the program /software has to be built. Then, all the gathered requirements are analysed to arrive at the scope or the objective of the overall software product. The last activity in this phase includes documenting every identified requirement of the users in order to avoid any doubts or uncertainty regarding the functionality of the programs.

Design The requirements documented in the previous phase acts as an input to the design phase. In the design phase, a plan of actions is made before the actual development process can start. This plan will be followed throughout the development process. Moreover, in the design phase, the core structure of the software program is broken down into modules. The solution of the program is then specified for each module in the form of algorithms or flowcharts. The design phase, therefore, specifies how software

Implementation In this phase, the designed algorithms are converted into program code using any of the high-level languages. The particular choice of language will depend on the type of program, such as whether it is a system or an application program. While is preferred for writing system programs, Visual Basic might be preferred for writing an application program. The program codes are tested by the programmer to ensure their correctness.

This phase is also called construction or code generation phase as the code of the software is generated in this phase.

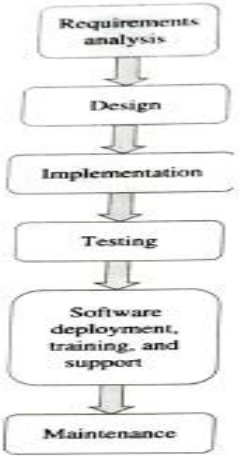
Testing In this phase, all the modules are tested together to ensure that the overall system works well as a whole product. Although individual pieces of codes are already tested by the programmers in the implementation phase, there is always a chance for bugs to creep into the program when the individual modules are integrated to form the overall program structure.

In this phase, the software is tested using a large number of varied inputs, also known as test data, to ensure

that the software is working as expected by the user's requirements that were identified in the requirements analysis phase.

Software deployment, training, and support After the code is tested and the software or the program has been approved by the users, it is installed or deployed in the production environment. This is a crucial phase that is often ignored by most developers. As a part of the deployment phase, it has become very crucial to have training classes for the users of the software.

Maintenance Maintenance and enhancements are ongoing activities that are done to cope with newly discovered problems or new requirements. Such activities may take a long time to complete as the requirement may call for the addition of new code that does not fit the original design or an extra piece of code, required to fix an unforeseen problem.



Phases in software development life cycle

3.4 PROGRAM DESIGN TOOLS: ALGORITHMS, FLOWCHARTS, PSEUDOCODES

3.4.1 Algorithms

The algorithm gives the logic of the program, that is, a step-by-step description of how to arrive at a solution.

In general term an algorithm provides a blueprint to writing a program to solve a particular problem.

In order to qualify as an algorithm, a sequence of instructions must possess the following characteristics:

- It should have finite number of steps
- Be precise
- Be unambiguous
- Not even a single instruction must be repeated infinitely.
- After the algorithm gets terminated, the desired result must be obtained.

Control Structures Used In Algorithms

An algorithm may employ three control structures, namely, sequence, decision, and repetition.

Sequence - Sequence means that each step of the algorithm is executed in the specified order.

Decision – These statements are used when the outcome of the process depends on some condition.

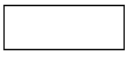
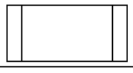
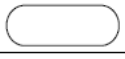
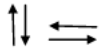
Repetition - Repetition, which involves executing one or more steps for a number of times, can be implemented using constructs such as the while, do-while, and for loops. These loops execute one or more steps until some condition is true.

3.4.2 Flowcharts

A flowchart is a graphical or symbolic representation of a process. It follows a top down approach in solving a problem.

Significance of flowcharts

- *It is drawn in the early stages of formulating computer solutions*
- *It facilitates communication between programmers and users*
- Programmers can make users understand the solution easily and clearly

Symbol	Name	Function
	Process	Indicates any type of internal operation inside the Processor or Memory
	input/output	Used for any Input / Output (I/O) operation. Indicates that the computer is to obtain data or output results
	Decision	Used to ask a question that can be answered in a binary format (Yes/No, True/False)
	Labelled Connector	Allows the flowchart to be drawn without intersecting lines or without a reverse flow.
	Predefined Process	Used to invoke a subroutine or an Interrupt program.
	Start/Stop	Indicates the starting or ending of the program, process, or interrupt program
	Flow Lines	Shows direction of flow.

Advantages

- They are very good communication tools to explain the logic of a system to all concerned.
- They are also used for program documentation.
- They act as guide or blueprint for the programmers.
- They direct the programmers to go from the starting point of the program to the ending point without missing any step in between.
- They can be used to debug programs that have errors.
- They help the programmers to easily detect, locate, and remove mistakes in the program in a systematic manner.

Limitations

- Drawing flowcharts is a laborious and a time-consuming activity.
- The flowchart of a complex program becomes complex and clumsy
- At times, a little bit of alteration in the solution may require complete redrawing of the flowchart
- The essentials of what is done may get lost in the technical details of how it is done,
- There are no well-defined standards that limit the details that must be incorporated into a flowchart

3.4.3 Pseudocodes

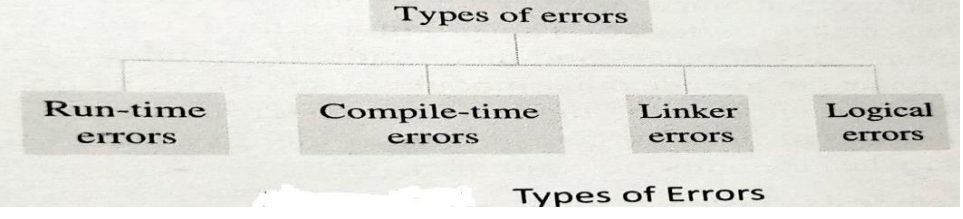
Pseudocode is a compact and informal high-level description of an algorithm that uses the structural conventions of a programming language. An ideal pseudocode must be complete, describing the entire logic of the algorithm, so that it can be translated straightaway into a programming language.

Features

- The sole purpose of pseudocodes is to enhance human understandability of the solution.
- They are commonly used in textbooks and scientific publications and for documenting algorithms, and for sketching out the program structure before the actual coding is done.
- This helps even non-programmers to understand the logic of the designed solution.
- There are no standards defined for writing a pseudocode, because a pseudocode is not an executable program. Flowcharts can be considered graphical alternatives to pseudocodes, but require more space on paper.

8.5 TYPES OF ERRORS

While writing programs, very often we get errors in our programs. These errors if not removed will either give erroneous output or will not let the compiler to compile the program.



Run-time Errors - Run-time errors occur when the program is being run executed. Such errors occur when the program performs some illegal operations like -

- dividing a number by zero
- opening a file that already exists
- lack of free memory space
- finding square root or logarithm of negative numbers

Run-time errors may terminate program execution, so the code must be written in such a way that it handles all sorts of unexpected errors rather terminating it unexpectedly.

Compile-time Errors - Compile- time errors occur at the time of compilation of the program.

Such errors can be further classified as follows:

Syntax Errors - Syntax errors are generated when rules of a programming language are violated.

Semantic Errors - Semantic errors are those errors which may comply with rules of the programming language but are not meaningful to the compiler.

Logical Errors - Logical errors are errors in the program code that result in unexpected and undesirable output which is obviously not correct. Such errors are not detected by the compiler, and programmers must check their code line by line or use a debugger to locate and rectify the errors.

Linker Errors - These errors occur when the linker is not able to find the function definition for a given prototype.

3.6 TESTING AND DEBUGGING APPROACHES

Testing is an activity that is performed to verify correct behavior of a program. It is specifically carried out with an intent to find errors. Ideally testing should be conducted at all stages of program development. However, in the implementation stage, the following three types of tests can be conducted

Unit Tests Unit testing is applied only on a single unit or module to ensure whether it exhibits the expected behavior.

Integration Tests These tests are a logical extension of unit tests. In this test, two units that have already been tested are combined into a component and the interface between them is tested. This process is repeated until all the modules are tested together. The main focus of integration testing is to identify errors that occur when the units are combined.

System Tests System testing checks the entire system. For example, if our program code consists of three modules then each of the module is tested individually using unit tests and then system test is applied to test this entire system as one system.

Debugging - It is an activity that includes execution testing and code correction. The main aim of debugging is locating errors in the program code. Once the errors are located, they are then isolated and fixed to produce an error-free code.

Different approaches applied for debugging a code includes:

Brute-Force Method In this technique, a printout of CPU registers and relevant memory locations is taken, studied, and documented. It is the least efficient way of debugging a program and is generally done when all the other methods fail.

Backtracking Method It is a popular technique that is widely used to debug small applications. It works by locating the first symptom of error and then tracing backward across the entire source code until the real cause of error is detected. However, the main drawback of this approach is that with increase in number of source code lines, the possible backward paths become too large to manage.

Algorithms, Flowcharts and Pseudocodes

1. Sum of 2 numbers

Algorithm

Step 1: [initialise]
Start

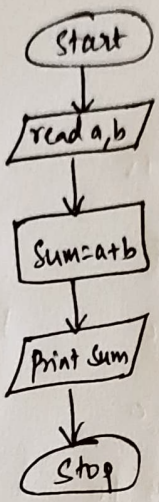
Step 2: [Input a, b]
read a, b

Step 3: [Calculate Sum = a+b]
Sum = a+b

Step 4: [Output sum value]
Print Sum

Step 5: [Terminate]
Stop

Flow chart



Pseudocode

1. Read a, b
2. Calculate Sum=a+b
3. print Sum
4. End

Variables: a, b, Sum

2. Leap year or not

Algorithm

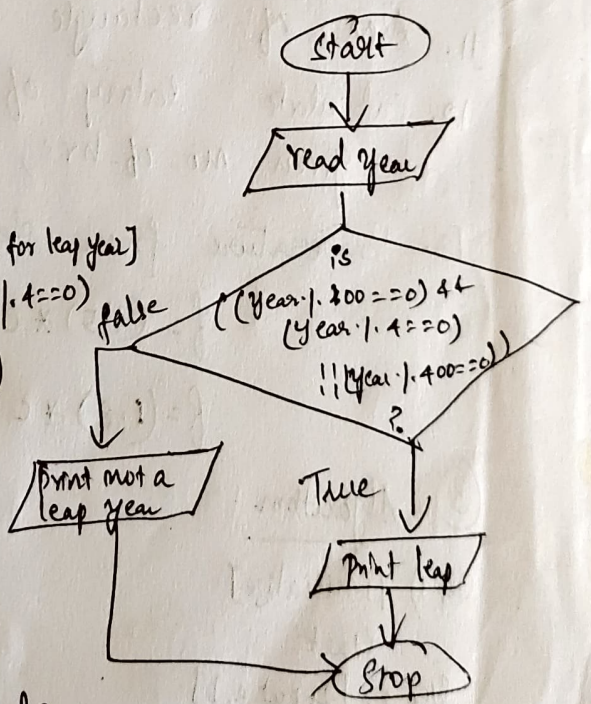
Step 1: [initialize]
Start

Step 2: [Input year]
read year

Step 3: if ((year .%. 100 != 0) && (year .%. 4 == 0)
|| (year .%. 400 == 0))
[perform the check for leap year]
print leap year
Else
print not leap year

Step 4: [Terminate]
Stop.

flowchart



Pseudocode

1. Read year
2. if ((year .%. 100 != 0) && (year .%. 4 == 0)
|| (year .%. 400 == 0))
Print leap year
Else print not leap year
Endif
3. End

Variables: year

5. Two numbers are equal or not
4. print 1 to 10
5. Swap 2 values
6. largest of 2 nos (a<b, a>b, a=b)
7. odd or even
8. grade

Above 75	O
60 – 75	A
50 – 60	B
40 – 50	C
less than 40	D

9. Sum of n natural numbers
10. Area of triangle ~~& rectangle~~.
11. Area of rectangle
12. Calculate salary of a daily wager given no_of_hrs, pay_per_hour, travel_allowance.
13. temperature from degree Celsius to fahrenheit.

$$f = (9/5) * c + 32 \quad (or)$$

$$f = (1.8) * c + 32$$

(3) Algorithm

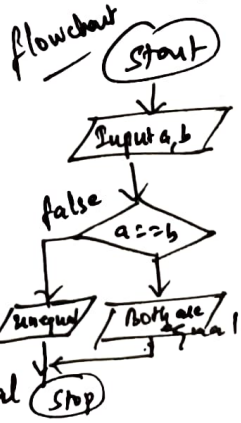
Step 1) [initialize]
start

Step 2 - [Input a, b]
read a, b

Step 3 - [check both nos are equal or not]
if (a==b)
print both are equal
else
print both are unequal

Step 4: [Terminate]
Stop

flowchart



Pseudocode

1) read a, b
2) if (a==b)
print equal
else
print unequal
endif
3) End
Variables - a, b

4. Print 1 to 10

Step 1: [initialize]
Start

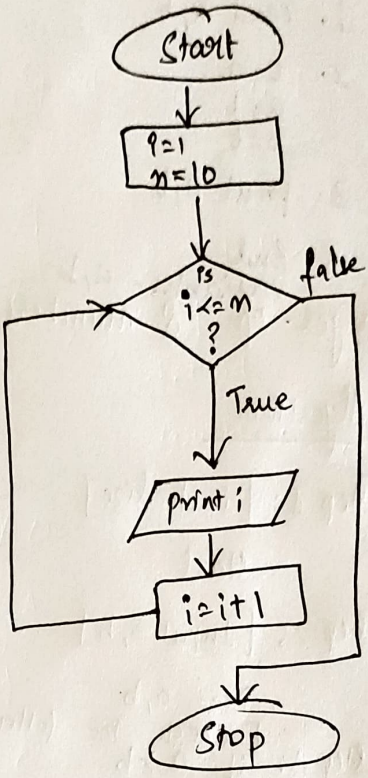
Step 2: [Set i, n]
i = 1
n = 10

Step 3: [Repeat steps 4, 5]
while i <= n
repeat steps 4, 5

Step 4: [print i]
output i

Step 5: [set i = i+1]
i = i+1

Step 6: [Terminate]
Stop



Pseudocode

1. Set i=1, n=10
2. while i <= n
   print i
   i = i+1
   End while
3. End

Variables: i, n

5. Swap 2 values

Algorithm

Step 1: [initialize]
Start

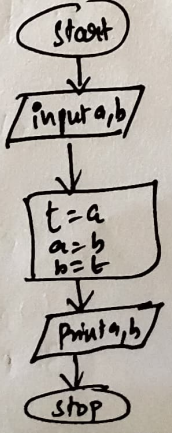
Step 2: [Input a, b]
read a, b

Step 3: [use temp variable t to swap]
t = a
a = b
b = t

Step 4: [Output a, b]
print a, b

Step 5: [Terminate]
Stop

Flowchart



Pseudocode

1. read a, b
2. t = a
   a = b
   b = t
3. print a, b
4. End

variables - a, b

6. larger of 2 numbers (a<b, a>b, a=b)

Algorithm

Step 1: [initialize]
Start

Step 2: [Input a, b]
read a, b

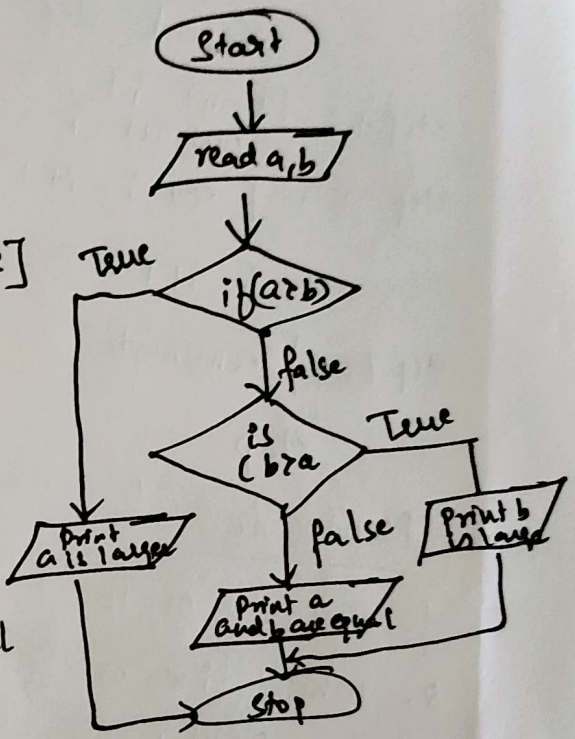
Step 3: [Check the following Conditions]
if a > b
print a is larger

Step 4: else if b > a
print b is larger

Step 5: else
print a and b are equal

Step 6: [Terminate]
Stop

Flowchart



~~Pseudocode~~ pseudocode

1. read a, b
2. if (a>b)
   print a is larger
   else if (a<b)
   print b is large
   else
   print a and b are equal
   endif
3. end

Variables: a, b

7. Odd or even

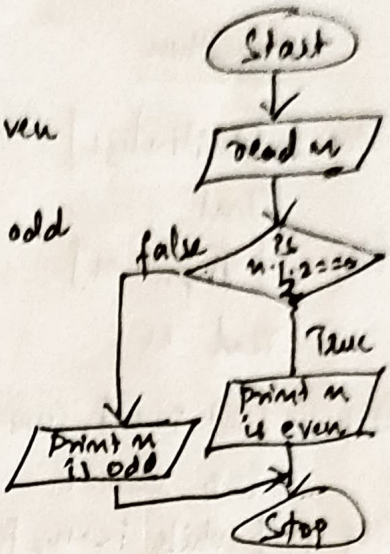
Algorithm

Step 1: [initialize]
Start
Step 2: [Input n]
read n
Step 3: [Check n.1.2==0]
if (n.1.2==0)
print n is even number
else
print n is odd number
Step 4: [Terminate]
Stop

Pseudocode

1. read n
2. if (n.1.2==0)
print n is even
else
print n is odd
endif
3. End
Variables - n

Flowchart



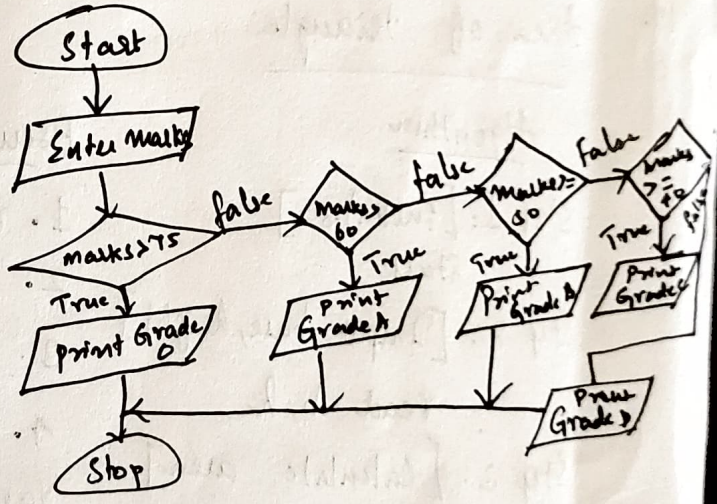
8. Grade

Above 75	O
60-75	A
50-60	B
40-50	C
less than 40	D

Algorithm

Step 1: [initialize]
Start
Step 2: [Input marks]
read marks
Step 3: [print the Grade]
if marks > 75
print Grade O
else if marks >=60
print Grade A
else if marks >=50
print Grade B
else if marks >= 40
Print Grade C
else
Print Grade D
Step 4: [Terminate]
Stop

flow chart



Pseudocode

1. read marks
2. if marks >=75
print Grade O
else if marks >=60
print Grade A
else if marks >=50
Print Grade B
else if marks >= 40
print Grade C
else
print Grade D
3. End.
Variables - marks

9. Sum of n natural numbers

Algorithm

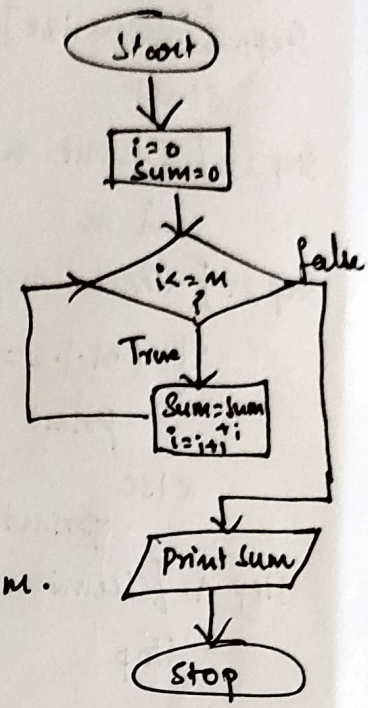
Step 1: [initialize]
Start
Step 2: [Input n]
read n
Step 3: [Set i=0], Sum=0
i=0, Sum=0
Step 4: [While i<=n, perform]
While (i<=n)
Sum = Sum + i
i=i+1
Step 5: [output Sum]
print Sum
Step 6: [Terminate]
Stop

pseudocode

1. read n
2. i=0
   Sum=0
3. while (i<=n)
   a. Sum=sum+i
   b. i=i+1
   Endwhile
4. print Sum
5. End

Variables - n, i, Sum.

flow-chart



10. Area of triangle

Algorithm

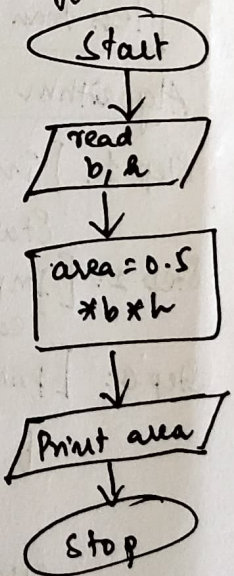
Step 1: [initialize.]
Start
Step 2: [Input base, height]
read b, h
Step 3: [Calculate area]
area = 0.5 * b * h
Step 4: [Output area]
print area.
Step 5: [Terminate]
Stop

Pseudocode

1. read b, h
2. area = 0.5 * b * h
3. print area
4. End

Variables - b, h, area

flow-chart



Area of rectangle

Algorithm

Step 1: [initialize]
Start

Step 2: [Input length, breadth]
read len, breadth

Step 3: [Calculate area]
area = len * breadth

Step 4: [Output area]
print area

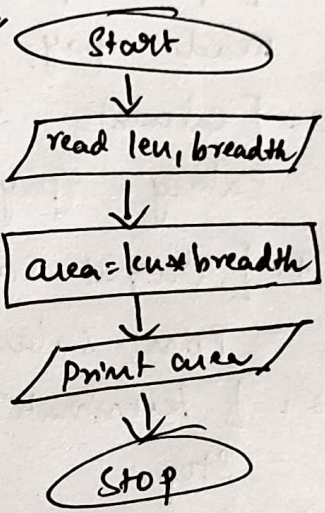
Step 5: [Terminate]
stop

Pseudocode

1. read len, breadth
2. area = len * breadth
3. print area
4. End

Variables - len, breadth, area

flowchart



13. temperature from degree celsius to fasheenheit

$f = (9/5) * C + 32$

Step 1: [initialize]
Start

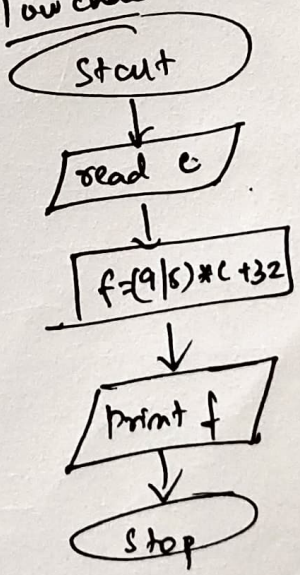
Step 2: [Input temp in celsius, as c]
read c

Step 3: [Calculate temp in fahrenheit]
$f = (9/5) * c + 32$

Step 4: [Output temp in fahrenheit]
print f

Step 5: [terminate]
stop

flow chart



Pseudocode

1. read c
2. f = (9/5) * +32
3. print f
4. End

Variables - c, f

12) Calculate salary of daily wager given no_of_hrs, pay_per_hour, travel_allowance.

Algorithm

Step 1: [initialize]
Start

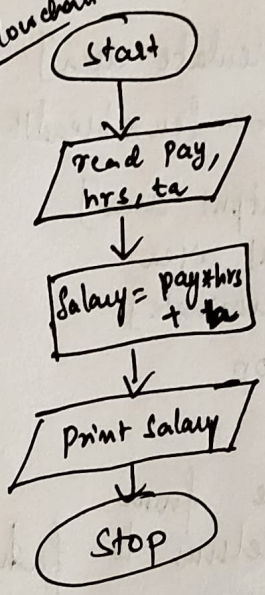
Step 2: [input pay_per_hour, no_of_hrs, travel_allowance]
read pay, hrs, ta

Step 3: [Calculate salary]
salary = (pay * hrs) + ta

Step 4: [Output salary]
print salary

Step 5: [Terminate]
Stop

flowchart



Pseudocode

1. read pay, hrs, ta
2. salary = (pay*hrs) + ta
3. print salary
4. End

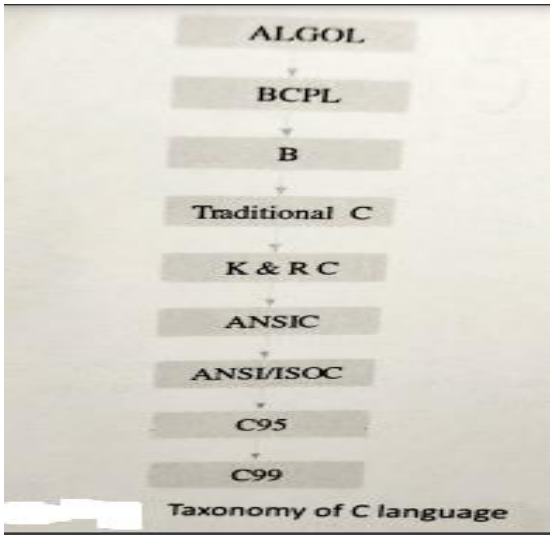
Variables — pay, hrs, ta

Chapter 4: Introduction to C

4.1 Introduction

The Programming language C was developed in the early 1970s by Dennis Ritchie at Bell Laboratories to be used by Unix Operating System. It was named 'C' because many of its features were derived from an earlier language called B. Although, C was designed for implementing system software, it was later on widely used for developing portable application software.

Background



Taxonomy of C language

1. ALGOL - Algorithmic Language uses block structure (1960)
2. BCPL - Basic Combined Programming Language (1967 – Martin Richards) – type-less, direct access of memory, so helped system programmers
3. B – (1970 Ken Thompson) Used to develop first version of UNIX.
4. Traditional C – (Dennis Ritchie 1972 at bell Laboratories) – ALGOL+BCPL+B+datatypes
5. K & R C – Traditional C was documented and popularized in the book 'The C Programming Language by Brian W. Kernighan and Dennis Ritchie in 1978.
6. Ansi C - The American National Standards Institute (ANSI) started working on defining the standard for C. This standard was approved in December 1989 and came to be known as ANSI C.
7. Ansi / ISO C - In 1990, the International Standards Organization (ISO) adopted the ANSI standard. This version of C came to be known as C89.
8. C95 - In 1995, some minor changes were made to C89; the new modified version was known as C95.
9. C99 - During 1990s, C++ and Java programming languages became popular among the users so the Standardization Committee of C felt that a few features of C++ / Java if added to C would enhance its usefulness. So, in 1999 when some significant changes were made to C95, the

modified version came to be known as C99.

Some of the changes made in the C99 version are as follows:

- Extension to character data types, so that they can support even non-English characters
- Boolean data type
- Extension to the integer data type
- Inclusion of type definitions in the for statement
- Inclusion of imaginary and complex types
- Addition of //, better known as C++ style line comment

Characteristics & Uses of C Language:

C is a robust language whose rich set of built-in functions and operators can be used to write complex programs.

The C compiler combines the features of assembly languages and high-level languages, which makes it best suited for writing system software as well as business packages.

- C is a high-level programming language, which enables the programmer to concentrate on the problem at hand and not worry about the machine code on which the program would be run.
- Small size - C has only 32 keywords. This makes it relatively easy to learn as compared to other languages.
- C makes extensive use of function calls.
- C is well suited for structured programming.
- Unlike PASCAL it supports loose typing (as a character can be treated as an integer and vice versa).
- Structured language as the code can be organized as a collection of one or more functions.
- Stable language. ANSI C was created in 1983 and since then it has not been revised.
- Quick language as a well written C program is likely to be as quick as or quicker than a program written in any other earlier language.
- Facilitates low level (bitwise) programming.
- Supports Pointers to refer computer memory, arrays, structures and functions
- C is a core language as many other programming languages (like C++, Java) are based on C
- If we know C, learning other computer languages becomes much easier.
- C is a portable language, i.e. a C program written for one computer can be run on another computer with little or no modification.
- C is an extensible language as it enables the user to add his own functions to the C library.
- C is often treated as the second best language for any given programming task.

4.2 Structure of a C Program

Documentation Section (optional)
Preprocessor Directives
Global Declarations
main() **{** **Local declarations** **Statements** **}**
function1() **{** **Local declarations** **Statements** **}** **…….**
functionn() **{** **Local declarations** **Statements** **}**

Structure of a C program

- A C program is composed of preprocessor commands, a global declaration section, and one or more functions.
- The preprocessor directives contain special instructions that indicate how to prepare the program for compilation.
- One of the most important and commonly used preprocessor commands is **include** which tells the compiler that to execute the program, some information is needed from the specified header file. Ex - **#include<stdio.h>, #define PI 3.14**
- Global declaration part will be revisited in the chapter on Functions.
- A program contains one or more functions, where a function is defined as a group of C statements that are executed together.
- The statements in a C function are written in a logical sequence to perform a specific task.
- The main() function is the most important function and is a part of every C program. The execution of a C program begins at this function.

- All functions (including main()) are divided into two parts the declaration section and the statement section.
- The declaration section precedes the statement section and is used to describe the data that will be used in the function.
- The data declared within a function are known as local declaration as that data will be visible only within that function, (i.e) the life-time of the data will be only till the function ends.
- The statement section in a function contains the code that manipulates the data to perform a specified task.
- From the structure we can conclude that a C program can have any number of functions depending on the tasks that have to be performed, and each function can have any number of statements arranged according to specific meaningful sequence.

Programmers can choose any name for functions. It is not mandatory to write Function1, Function2, etc., but with an exception that every program must contain one function that has its name as main().

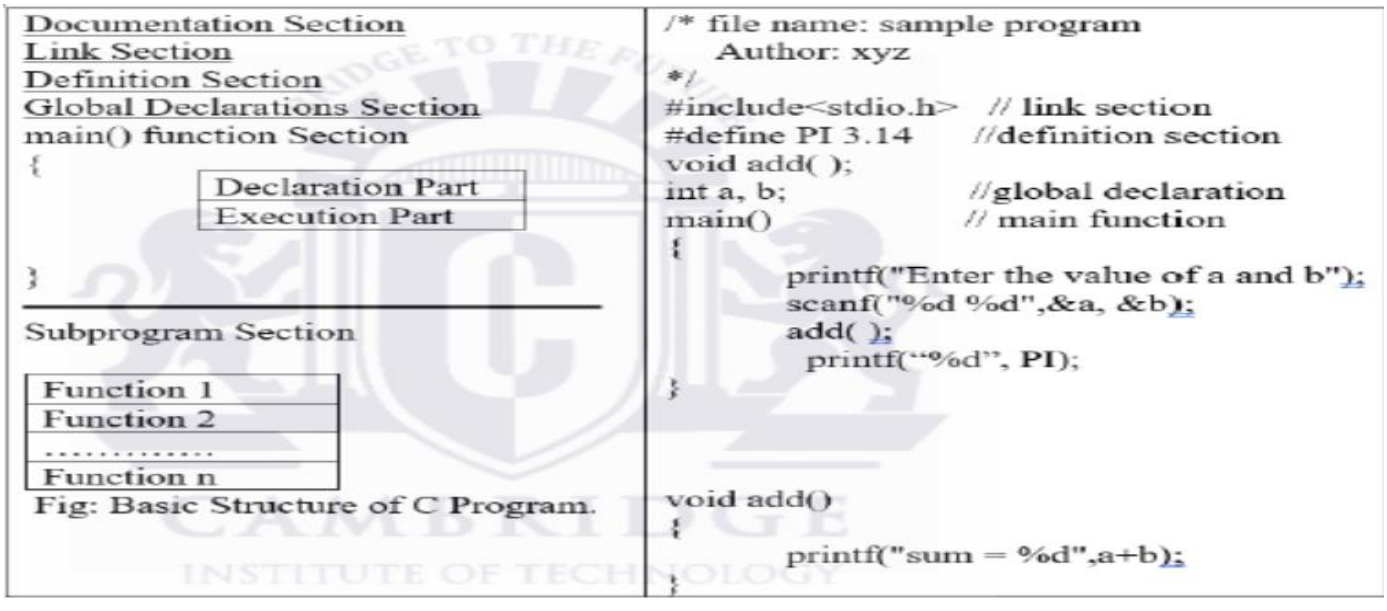


Fig: Basic Structure of C Program.

4.3 WRITING THE FIRST C PROGRAM

To write a C program, we first need to write the code. For this, open a text editor. If we are a Windows user we may use Notepad and if we prefer working on UNIX/Linux we can use emacs or vi editor. Once the text editor is opened on our screen, type the following statements

```
#include <stdio.h>
int main()
{
    printf("\n Welcome to the world of C");
    return 0;
```

}

Output

Welcome to the world of C

#include <stdio.h>

- This is a preprocessor command that comes as the first statement in our code.
- All preprocessor commands start with symbol hash (#).
- The include statement tells the compiler to include the standard input/output library or header file (stdio.h) in the program. This file has some in-built functions.
- By simply including this file in our code we can use these functions directly.
- The standard input output header file contains functions for input and output of data as reading values from the keyboard and printing the results on the screen.

int main()

- Every C program contains a main() function which is the starting point of the program.
- int is the return value of the main() function.
- After all the statements in the program have been written, the last statement of the program will return an integer value to the operating system.

{} The two curly brackets

- These are used to group all the related statements of the main function. All the statements between the braces form the function body. The function body contains a set of instructions to perform the given task.

printf("\n Welcome to the world of C");

- The printf function is defined in the stdio.h file and is used to print text on the screen.
- The message that has to be displayed on the screen is enclosed within double quotes and put inside brackets.
- The message is quoted because in C a text (also known as a string between characters) is always put between inverted commas.
- \n is an escape sequence and represents a newline character.
- It is used to print the message on a new line on the screen.
- Like the newline character, the other escape sequences supported by C language :

Escape sequences

Escape sequence	Purpose	Escape sequence	Purpose
\a	Audible signal	\?	Question mark
\b	Backspace	\\	Back slash
\t	Tab	\'	Single quote
\n	Newline	\"	Double quote
\v	Vertical tab	\0	Octal constant
\f	New page\ Clear screen	\x	Hexadecimal constant
\r	Carriage return		

If we do not place a parenthesis after main, a compiler error will be generated.
Placing a semicolon after the parenthesis of main function will generate a compiler error.
Escape sequences are actually non-printing control characters that begin with a backslash (\)
Every statement in the main function ends with a semi-colon (;)

return 0;

- This is a return command that is used to return the value 0 to the operating system to give an indication that there were no errors during the execution of the program.
- We have written all the statements using the text editor, save the text file as hello.c.
- If we are a Windows user then open the command prompt by clicking Start->Run and typing 'command' and clicking ok.
- Using the command prompt, change to the directory in which we had saved our file and then type: **tc hello.c**
- In case if we are working on UNIX/Linux operating system, then exit the text editor and type **cc hello.c –o hello**
  The -o is for the output file name. If we leave out the –o - then the file name a.out is used
  This command is used to compile our C Program.
- If there are mistakes in the program then the compiler will tell us the mistake we have made and on which line we made it.
- In case of errors we need to re-open our .C file and correct those mistakes.
- However, if everything is right then no error(s) will be reported and the compiler will

create an exe file for our program.

- This .exe file can be directly run by typing 'hello.exe' for Windows and ‘./hello' for UNIX/ Linux operating system. When we run the .exe file, the output of the program will be displayed on screen. That is, Welcome to the world of C

Note :
The printf and return statements have been indented or moved away from the left side.
This is done to make the code more readable.

➢ **Using Comments**

Comments are just a way of explaining what a program does. The compiler ignores comments when forming the object file. Comments are non-executable statements. Commented statements can be used anywhere in the program.

C supports 2 types of comments:

1) Line comment : // is used to comment a single statement. A line comment can be placed anywhere on the line and it does not require to be ended.
2) Block comment : /*is used to comment multiple statements. A /* is ended with */ and the sattements that lie within these characters are commented.

```
/* Author : Sonia
File name : hello.c
Description : to print ‘welcome to the world of C’ on the screen */
#include <stdio.h>
int main()
{
        // Prints message
        printf("\n Welcome to the world of C");
        return 0; // returns a value 0 to the OS
}
```

Output

```
Welcome to the world of C
```

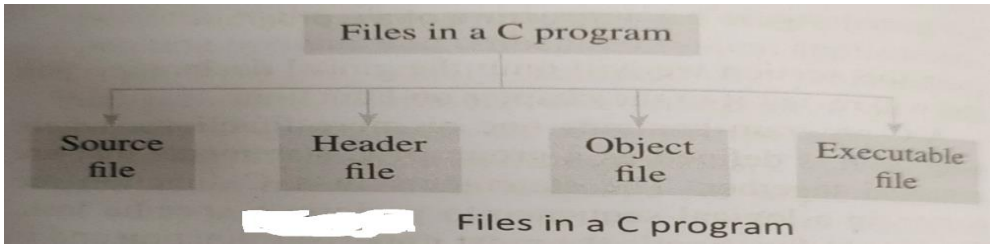
4.4 FILES USED IN A C PROGRAM

Every C Program has four kinds of files associated with it. These include:

4.4.1 Source Code Files

The source code file contains the source code of the program. The file extension of any C source code file is '.c'. This file contains C source code that defines the main function and may be other

functions. The main() function is the starting point of execution when we successfully compile and run the program. Ex – hello.c



4.4.2 Header Files

When working with large projects, it is often desirable to separate out certain subroutines from the main() function of the program and store them in a different file known as header file.

The advantages of header files can be realized in the following cases:

- The programmer wants to use the same subroutines in different programs.
- The programmer wants to change or add subroutines and have those changes reflected in all the other programs.

Header files names ends with a '.h' extension

Although some standard header files are automatically available to C programmers, in addition to those header files, the programmer may have his own **user-defined header files**.

- Standard Header Files - In the program, we used printf() function that has not been written by us. We do not know the details of how this function works. Such functions that are provided by all C compilers are included in standard header files.
- Examples of these standard header files include:

  string.h: for string handling functions

  stdlib.h: for some miscellaneous functions

  stdio.h: for standardized input and output functions

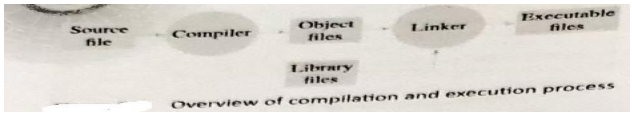
  math.h: for mathematical functions

  alloc.h: for dynamic memory allocation

  conio.h: for clearing the screen:
- All the header files are referenced at the start of the source code file that uses one or more functions from that file.

4.4.3 Object Files

Compiler - A compiler is a special program that translates a programming language's source code into object code.



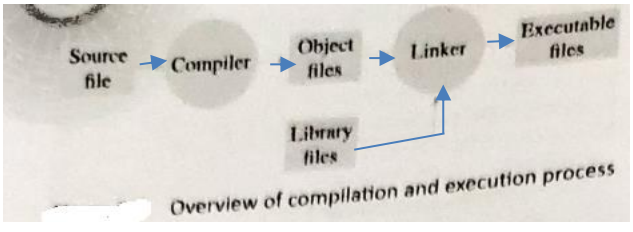
Compilation - The compilation is the process of transforming source code into object code. Object files are generated by the compiler as a result of processing the source code file. Object files have a '.o' extension, although some operating systems including Windows and MS-DOS have a '.obj' extension for the object file.

4.4.4 Binary Executable Files

The binary executable file is generated by the linker. The linker links the various object files to produce a binary file that can be directly executed. On Windows operating system, the executable files have a '.exe' extension.

4.5 COMPILING AND EXECUTING C PROGRAMS

- C is a compiled language. So once a C program is written, we must run it through a C compiler that can create an executable file to be run by the computer.
- While the C program is human-readable, the executable file, on the other hand, is a machine-readable file available in an executable form.
- **Every programming language has its own compiler.**
  The compiler translates the source code into an object code. The object code contains the machine instructions for the CPU, and calls to the operating system. However, even the object file is not an executable file.
- Therefore, in the next step, ***the object file is processed with another special program called a linker. The output of the linker is an executable or runnable file.***



Overview of compilation and execution process

- *In C language programs, there are two kinds of source files. In addition to the main (c) source file, which contains executable statements there are also header (.h) source files.*
- *Since all input and output in C programs is done through library functions, every C program therefore uses standard header files.*
- *So, preprocessing is done by the pre-processor program before the actual compilation.*
- *The preprocessor reads the source file as text, and produces another text file as output.*
- *Source code lines which begin with # symbol are written in preprocessor language.*
- *The output of a preprocessor is a text file which does not contain any preprocessor statements. This file is then processed by compiler and then by linker to produce the final executable file.*

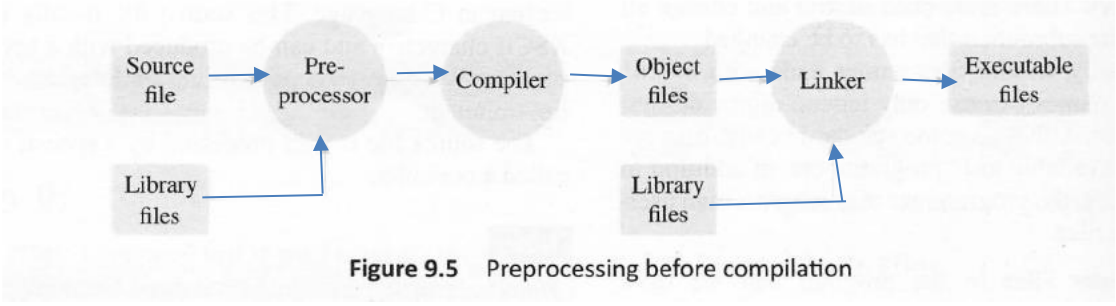


Figure 9.5 Preprocessing before compilation

- In modular programming, the source code is divided into two or more source files. All these source files are compiled separately thereby producing multiple object files. These object files are then combined by the linker to produce an executable file.

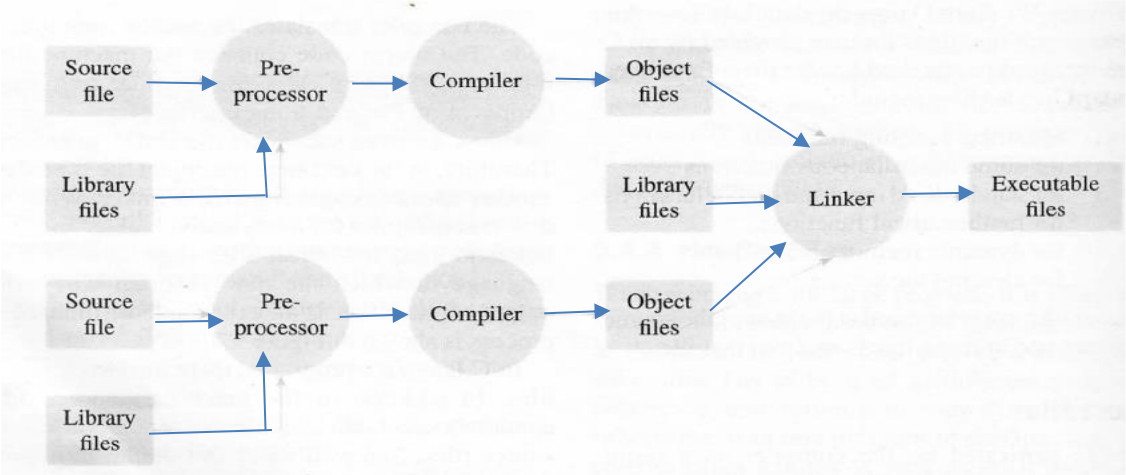
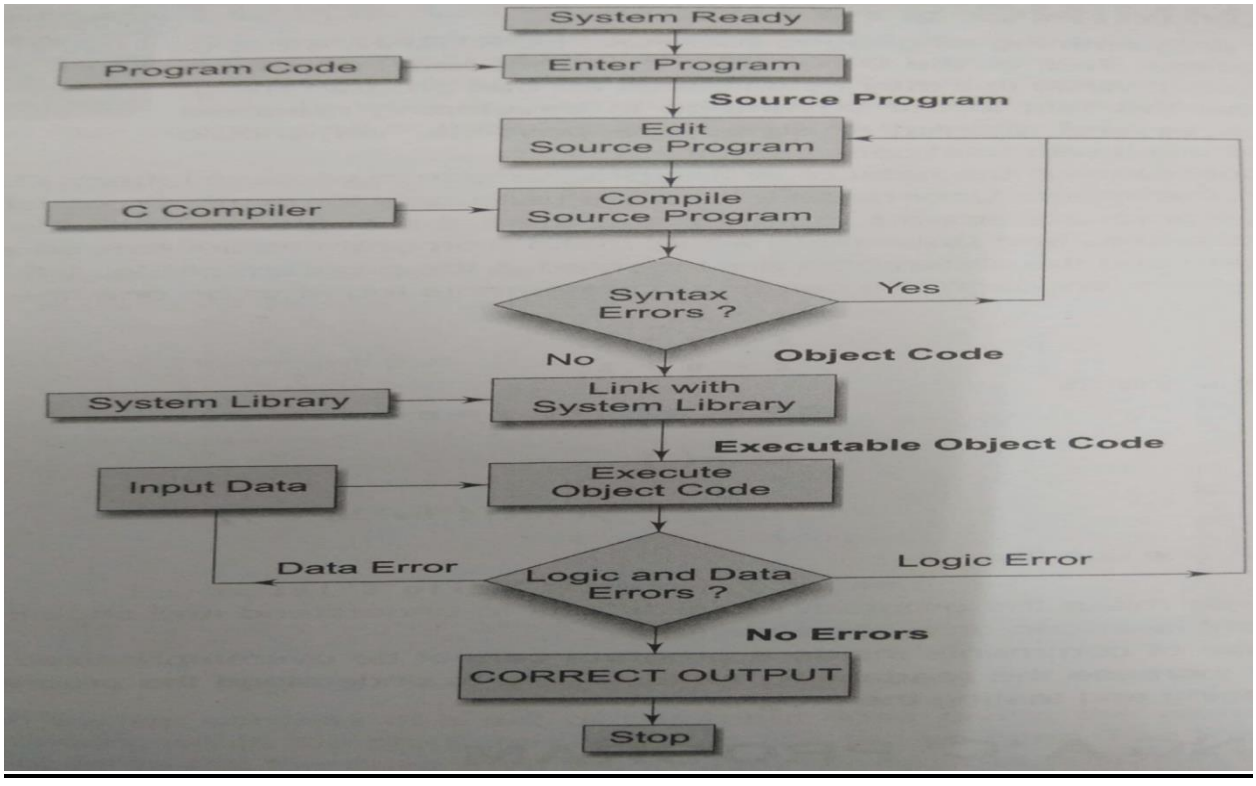


Figure 9.6 Modular programming—the complete compilation and execution process



4.6 Character Set in C

Like in natural languages, computer languages also use a character set that defines the fundamental units used to represent information. In C, a character means any letter from English alphabet, a digit or a special symbol used to represent information. These characters when combined together form tokens that act as basic building blocks of a C program.

The character set of C can therefore be given as:

a. English alphabet: Include both lower case (a - z) as well as upper case (A - Z) letters

b. Digits: Include numerical digits from o to 9

c.

Special characters: Include symbols such as ~, @, %, ^, &, *,{, }, <, >, =, _, +, -, $, /, (,), \, ;, :, [,], ‘, ", ?, ., !, |

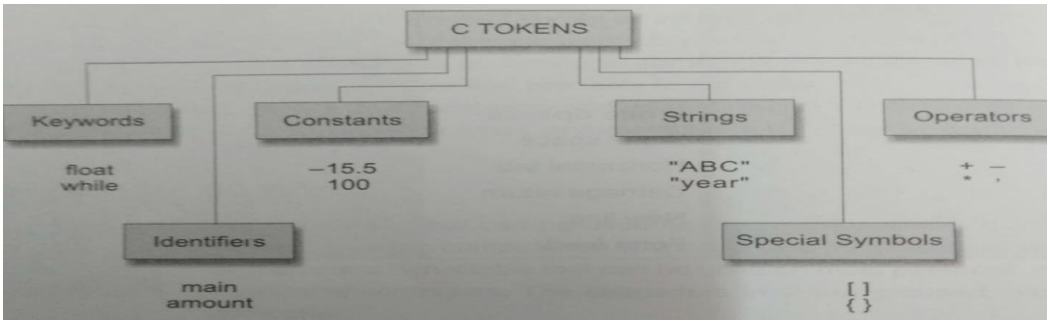
d. White space characters: These characters are used to print a blank space (space) on the screen.

Ex- \t horizontal tab, \n newline, \v vertical tab

e. Escape sequence: Ex- \n,\t,\r,\a,\b

4.7 C Tokens

- Tokens are the basic building blocks in C language.
- It is the smallest individual unit in a C program.
- Individual word which has a meaning are called tokens.
- There are six types of tokens in C.



4.8 KEYWORDS

Like every computer language, C has a set of reserved words often known as keywords that cannot be used as an identifier. All keywords are basically a sequence of characters that have a fixed meaning. By convention all keywords must be written in lowercase (small) letters.

Table 9.2 Keywords in C language

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

4.9 Identifiers

- Identifiers help us to identify data and other objects in the program.
- Identifiers are basically the names given to program elements such as variables, arrays, and functions.
- Identifiers may consist of sequence of letters, numerals, or underscores.

Rules for Forming Identifier Names

Some rules have to be followed while forming identifier names. They are as follows:

- Identifiers cannot include any special characters or punctuation marks (like #, $, ^ , ? , . , etc.) except the underscore '_ ' .
- Keywords cannot be used as identifiers.
- The case of alphabetic characters that form the identifier name is significant. For example, 'FIRST' is different from 'first' and 'First'.
- Identifiers must begin with a letter or an underscore.
- Identifiers can be of any reasonable length containing letters, digits and underscores. Only first 31 characters are significant.

Examples of valid identifiers include: roll_number, marks, name, emp_number, basic_pay, HRA, DA, dept_code, DeptCode, RollNo, EMP_NO

Examples of invalid identifiers include: 23_student, %marks, @name, #emp_number, basic.pay, -HRA, (DA), &dept_code, auto

C is a case-sensitive language. Therefore rno, Rno, RNo, RNO are considered as different identifiers. If we type printf function as Printf, then an error will be generated.

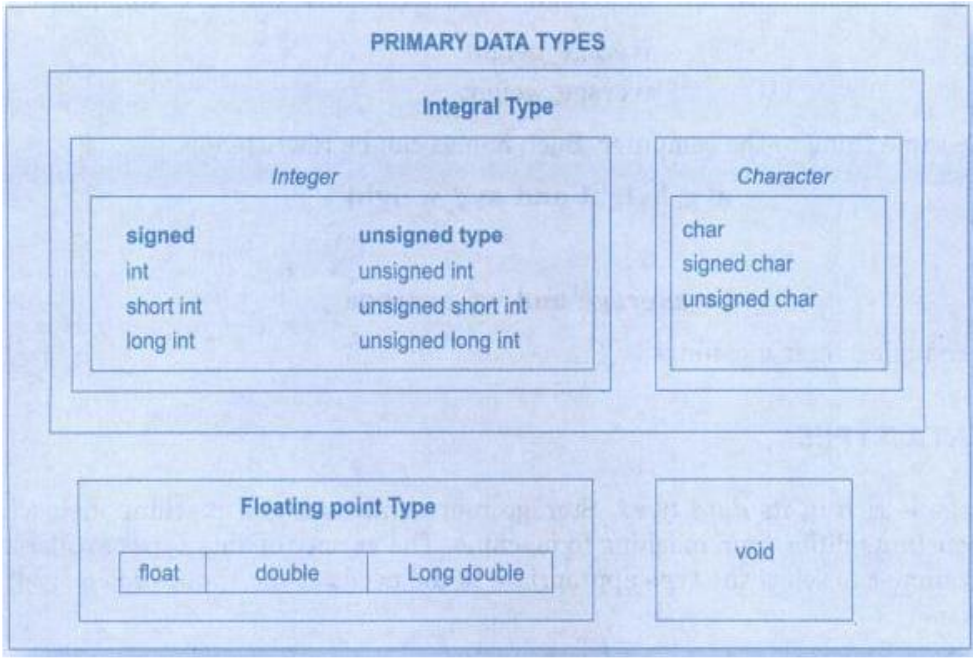
4.10 Basic Data Types in C

Data type is nothing but the type of data that we use to store a value in a variable. ANSI C supports three classes of data types.

i) Primary (or fundamental) data types ii) Derived data types iii) User-defined data types

Primary Data Types (Basic data types or Fundamental data types or Primitive data types)

All C compilers support five fundamental data types, namely integer (**int**), character (**char**), floating point (**float**), double-precision floating point (**double**) and **void**. In addition, C also supports four modifiers – two sign specifiers (signed and unsigned) and two size specifiers (short and long).



Size and Range of Basics Data Types on 16–bit Machines

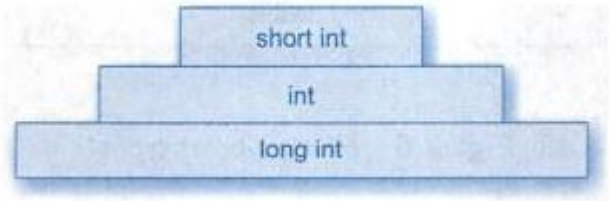
Table 9.3 Basic data types in C

Data type	Keyword used	Size in bytes	Range	Use
Character	char	1	−128 to 127	To store characters
Integer	int	2	−32768 to 32767	To store integer numbers
Floating point	float	4	3.4E−38 to 3.4E+38	To store floating point numbers
Double	double	8	1.7E−308 to 1.7E+308	To store big floating point numbers
Valueless	void	0	Valueless	—

Integer Types

Integers are whole numbers with a range of value supported by a particular machine.

If we use a 16 bit word length, the size of the integer value is limited to range (-32768 to +32767) -2^{15} to $+2^{15}-1$. The signed integer uses one bit for sign and 15 bits for the magnitude of the number.



The C has three classes of integer storage, namely **short int, int**, and **long int,** in both signed and unsigned forms. **short int** represents fairly small integer values and require half the amount of storage as a regular **int** number uses. The unsigned integers use all the bits for the magnitude of the number and are always positive. The range of unsigned integer number will be from 0 to 65,535. sign bit is the leftmost bit of a memory word which is used to determine the sign of the content stored in that word. When it is 0, the value is positive and when it is 1, the value is negative. When we do not specify the signed/unsigned modifier, C language automatically takes it as a signed variable.

Ex – int a;

Floating Point Types

Floating point (or real) numbers are stored in 32 bits (on 16 and 32 bit machines), with 6 digits of precision. Floating point numbers are defined in C by keyword float. To get more accuracy the type double can be used to define the number. A double data type number uses 64 bits giving a precision of 14 digits. These are known as double precision numbers.

Ex –

float a;

double b;

Detailed List of Data types		
Data type	Size in bytes	Range
char	1	-128 to +127 (-2^7 to $+2^7-1$)
unsigned char	1	0 to 255 (0 to 2^8-1)
signed char	1	-128 to +127 (-2^7 to $+2^7-1$)
int	2	-32768 to +32767 (-2^{15} to $+2^{15}-1$)
unsigned int	2	0 to 65535 (0 to $2^{16}-1$)
signed int	2	-32768 to +32767 (-2^{15} to $+2^{15}-1$)
short int	2	-32768 to +32767 (-2^{15} to $+2^{15}-1$)
unsigned short int	2	0 to 65535 (0 to $2^{16}-1$)
signed short int	2	-32768 to +32767 (-2^{15} to $+2^{15}-1$)
long int	4	-2147483648 to +2147483647 (-2^{31} to $+2^{31}-1$)
unsigned long int	4	0 to 4294967295 (0 to $+2^{32}-1$)

signed long int	4	-2147483648 to +2147483647 (-2^{31} to $+2^{31}-1$)
float	4	3.4E-38 to 3.4E+38
double	8	1.7E-308 to 1.7E+308
long double	10	3.4E-4932 to 1.1E+4932

Character Types

The char data type is of one byte and is used to store single characters. C does not provide any data type for storing text. This is because text is made up of individual characters. char is supposed to store characters not numbers, so why this range (-128 to 127)? The answer is that in memory characters are stored in their ASCII codes. For example, the character 'A' has the ASCII code 65. In memory we will not store 'A' but 65 (in binary number format).

Ex – char a;

Void Type

Last data type called void type holds no value. It is primarily used in three cases:

- To specify the return type of a function (when the function returns no value).
- To specify the parameters of the function (when the function accepts no arguments from the caller).
- To create generic pointers.

Derived Data types:

These are the data types that are derived from basic data types. Eg- arrays, pointers.

int a[10];

User defined data types:

1) In these data types, user defines an identifier for the existing data type using typedef feature.

Ex – typedef datatype identifier ;

typedef int integer;

integer a;

2) Structure in c is a user-defined data type that enables us to store the collection of different data types.

4.11 VARIABLES

- A variable is defined as a meaningful name given to a data storage location in computer memory.
- When using a variable, we actually refer to address of the memory where the data is stored.
- Unlike constants that remain unchanged during the execution of a program, a variable may take different values at different times during execution.

Examples: Averages, height, Total, Counter_1,.etc

Variable names (are nothing but identifiers) may consist of letters, digits, and the underscore(_) character, subject to the same conditions for identifiers.

C language supports two basic kinds of variables - numeric and character.

Numeric Variables – Variables that store numeric datatype (i.e) int and float.

Numeric variables can be used to store either integer values or floating point values. While an integer value is a whole number without a fraction part or decimal point, a floating point value can have a decimal point. Ex – int a=56;

Character Variables - Variables that store character datatype (i.e) char

Character variables are just single characters enclosed within single quotes. These characters could be any character from the ASCII character set-letters (‘a’ , ‘A’), numerals (‘2’), or special characters (‘&’). In C, a number that is given in single quotes is not the same as a number without them. This is because 2 is treated as an integer value but ‘ 2’ is a considered character not an integer. Ex- char a=’5’;

Declaring Variables

Each variable to be used in the program must be declared.

To declare a variable, specify the data type of the variable followed by its name.

Variable names should always be meaningful and must reflect the purpose of their usage in the program.

The memory location of the variable is of importance to the compiler only and not to the programmer.

In C, variable declaration always ends with a semicolon.

After designing suitable variable names, we must declare them to the compiler.

data-type v1,v2,v3,….vn;

Declaration does two things :

- It tells the compiler what the variable name is.
- It specifies what type of data the variable will hold.

for example:

```
int emp_num;
float salary;
```

In C variables must be declared before using them.

C allows multiple variables of the same type to be declared in one statement. So the following statement is legal in C.

```
float temp_in_celsius, temp_in_farenheit;
```

In C variables are declared at three basic places as follows:

- When a variable is declared inside a function it is known as a local variable.
- When a variable is declared in the definition of function parameters it is known as a formal parameters

- When the variable is declared outside all functions, it is known as a global variable.

A variable cannot be of type void.

Initializing Variables

The process of giving initial values to variables is called initialization. Assigning the values at the time of variable is declared this is done with the following form:

data_type variable= constant;

For example,

int emp_num = 7;

float salary = 2156.35;

char grade = 'A';

double balance_amount = 100000000;

The initializer applies only to the variable defined immediately before it. Therefore, the statement **int count, flag = 1;** initializes the variable flag and not count. If we want both the variables to be declared in a single statement then write, **int count = 0, flag = 1;**

When variables are declared but not initialized they usually contain garbage values.

4.12 CONSTANTS (Literals)

Constants are identifiers whose values do not change. While values of variables can be changed at any time, values of constants can never be changed.

The value of the constant is known to the compiler at the compile time.

C allows the programmer to specify constants of integer type, floating Point type, character type, and string type.

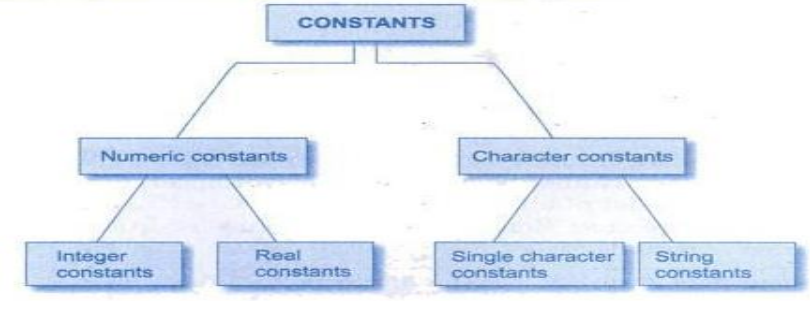


Fig Basic types of constants

Integer Constants

An integer constant refers to a sequence of digits. There are three types of integers, namely,

- Decimal integer (0 to 9) – 10 values
- Octal integer (0 to 7) – 8 values
- Hexadecimal integer (0 -9, A-F) -16 values

Decimal integers

Decimal integer consist of set of digits, 0 to 9, preceded by an optional – or + sign.

Examples:

123 -321 0 654321 +78

Embedded spaces, commas, and non-digit characters are not permitted between digits.

Example: 15 750 20,000 $1000 are illegal numbers.

Octal integers

An octal integer constant consists of any combination of digits from the set 0 through 7, with a leading 0(Zero). Some examples of octal integers are:

037 0435 0551

Hexadecimal integers

A sequence of digits preceded by 0x or 0X is considered as a hexadecimal integer. It includes digits from 0 through 9. They may also include alphabets A through F or a through f. The letter A through F represents the numbers 10 through 15. Following are the examples of valid hex integers:

0X2 0x9F 0Xbcd

We rarely use octal and hexadecimal numbers in programming.

It is also possible to store large integer constants on these machines by appending qualifiers such as U, L and UL to the constants. Examples:

56789U	or 56789u	(unsigned integer)
987612347UL	or 987612347ul	(unsigned long integer)
9876543L	or 9876543l	(long integer)

Real Constants (Floating Point Constants)

Integer numbers are inadequate to represent quantities that vary continuously, such as distances, heights, temperatures, prices and so on. These quantities are represented by numbers containing fractional parts like 17.548. Such numbers are called *real* (or *floating point*) constants.

Examples of real constants are: 0.0083 -0.75 435.36 +247.0

A real number may also be expressed in *exponential* (or *scientific*) notation. For example, the value 215.65 may be written as 2.1565e2 in exponential notation. E2 means multiply by 10^2. The general form is: **mantissa e exponent**

The *mantissa* is either a real number expressed in *decimal notation* or an integer. The *exponent* is an integer number with an optional *plus* or *minus* sign. The letter **e** separating the mantissa and the exponent can be written in either lowercase or uppercase.

0.65e4 12e-2 15e+5 3.18E3 -1.2E-1

Embedded white space is not allowed.

Exponential notation is useful for representing numbers that are either very large or very small in

magnitude. For example, 7500000000 may be written as 7.5E9 or 75E8. Similarly, -0.000000368 is equivalent to -3.68E-7.

Table : Examples of Numeric Constants

Constant	Valid?	Remarks
698354L	Yes	Represents long integer
25,000	No	Comma is not allowed
3.5e-5	Yes	
1.5E+2.5	No	Exponent must be an integer
$265	No	$ symbol is not permitted
0X7B	Yes	Hexadecimal integer

Single Character Constants:

A single character constant (or simply character constant) contains a single character enclosed within a pair of *single* quote marks. Example of character constants are:

'5' 'X' ';' ' '

Note that the character constant '5' is not the same as the number 5. The last constant is a blank space. For example, 'a' and ' @' are character constants. In computers, characters are stored using machine's character set using ASCII codes.

String Constants:-

A string constant is a sequence of characters enclosed in double quotes. The characters may be letters, numbers, special characters and blank space. Examples are:

"Hello" "1987" "WELL DONE" "?...!" "5+3" "X"

"a" is not the same as 'a' . The characters comprising the string constant are stored in successive memory locations. When a string constant is encountered in a C program, the compiler records the address of the first character and appends a null character ('\0') to the string to mark the end of the string. Thus, length of a string constant is equal to number of characters in the string plus 1 (for the null character). Therefore, the length of string literal "hello" is 6.

Declaring Constants

To declare a constant, precede the normal variable declaration with const keyword and assign it a value. For example, **const float pi = 3.14;**

Syntax: const datatype varname=constant;

The const keyword specifies that the value of pi cannot change.

However, another way to designate a constant is to use the pre-processor command define. Like other preprocessor commands, define is preceded with a # symbol.

Ex - **#define PI 3.14159**

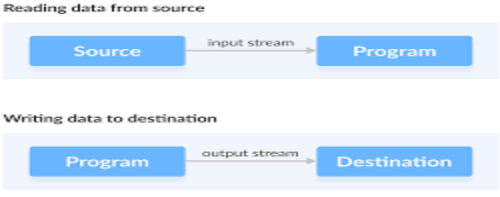
#define service_tax 0.12

- Constant names are written in CAPITALS to distinguish them from normal variable names.
- '#' must be the first character in the line.
- A blank space is required between **#define** and symbolic name and between the **symbolic name** and the constant.
- **#define** statements must not end with a semicolon.
- **After** definition, the symbolic name should not be assigned any other value within the program by using an assignment statement. **Example**: PI= 3.142; is illegal.

4.13 INPUT/OUTPUT STATEMENTS IN C

Streams

A stream is the source of data as well as the destination of data. Streams are associated with a physical device such as a monitor or a file stored on the secondary memory. We can do input/output from the keyboard/monitor or from any file.



1. One method of inputting is to assign values to variables is through the assignment statement such as x=5;
2. Another method is to use the input function **scanf** which can read data from a keyboard.

For outputting result **printf** is used.

All input/output operations are carried out through function calls such as printf and scanf.

These input/output operations are included in a statement: **#include<stdio.h>**

The file name stdio.h is an abbreviation for standard input-output header file. The instruction #include<stdio.h> tells the compiler to search for a file named stdio.h and place its contents at this point in the program. The contents of the header file become part of the source code when it is compiled.

Reading and writing can be done by **formatted and unformatted Input and output functions**.

Formatted I/O functions - use format specifiers. Can read and write any type of data like integers, characters, real numbers and strings.

Code	Meaning
%c	single character
%d	decimal integer
%o	octal integer

%x, %X	hexadecimal integer
%i	decimal, octal or hexadecimal integer
%e, %E	floating point value in exponential form
%f	floating point value in fractional form
%g, %G	floating point value in exponential or fractional
%h	short integer
%s	string
%u	unsigned decimal integer

Commonly used format codes

The following letters may be used as prefix for certain conversion characters

h for short integers

l for long integers or double

L for long double

Formatted Input: scanf() function is used.

The general form of **scanf** is

scanf (“control string”, arg1,arg2,…argn**);**

The control string specifies the type and format of the data that has to be obtained from the keyboard and stored in the memory locations pointed by arguments arg1,arg2….argn (i.e) the arguments are actually the variable addresses where each piece of data is to be stored. Control string and arguments are separated by comma.

scanf (“%d %c %f”, &x,&y,&z);

where x is a integer variable, y is a character and z is a float value.

Formatted input refers to an input data that has been arranged particular format.

For example: 15.75 123 John

This line contains three pieces of data, arranged in a particular form. For example, the first part of the data should be read into a variable float, the second into int, and the third part into char.

The control/format string contains format specifications, consisting of the conversion character %, a data type character and an optional number specifying the field width.

Inputting integer numbers

The field specification for reading an integer number is:

%wd

The percentage sign (%) indicates that a conversion follows. **w** is an integer number that specifies the field width of the number to be read and **d**, known as data type character, indicates that the number to be read is in integer mode. Consider the following example:

1. **scanf("%2d %5d", &num1,&num2);**

50 31426

The value 50 is assigned to num1 and 31426 to num2. Suppose if the input data line is **31426 50**, then 31 is assigned to num1 and num2 will be assigned 426. The value 50 is not read at all.

2. **scanf("%d %d", &num1,&num2);**

will read the data 31426 50 correctly and assign 31426 to num1 and 50 to num2.

An input field may be skipped by specifying *in the place of field width. For example, the statement

3. **scanf("%d %*d %d", &a,&b)**

will assign the data 123 456 789 as follows:

123 to a

456 skipped (because of *)

789 to b

The data type character **d** may be preceded by **'l'**(letter ell)to read long integers and **h** to read short integers.

4. **scanf("%ld %hd", &num1,&num2);**
5. **scanf("%d-%d", &num1,&num2);**

   123-456

   a=123

   b=456
6. **scanf("%d/%d", &num1,&num2);**

   123/456

   a=123

   b=456

INPUTTING REAL NUMBERS

For example, the statement : **scanf ("%f %f %f", &x,&y,&z);** with the input data

475.89 43.21E-1 678

will assign the value 475.89 to **x** , 4.321 to **y**, and 678.0 to **z**.

If the number to be read is of **double** type, then the specification should be **%lf.**

scanf ("%lf ", &x);

INPUTTING CHARACTER STRINGS

%c is used for inputting one character

A **scanf** function can input strings containing more than one character. Following are the specifications for reading character strings: **%ws or %wc or %s**

%c may be used to read a single character. **scanf ("%c ", &x);**

%s is used to read strings, the corresponding argument should be a pointer to a character array.

scanf ("%s ", x); Here x is a character array.

1) **char a;**
   scanf("%c",&a);
2) **char a[100];**
   scanf("%s",a);
3) **char a[100];**
   scanf("%5c",a);
4) **char a[100];**
   scanf("%5s",a);
   The last character by default in a string is null character ('\0').

Reading mixed data types

The statement

scanf ("%d %c %f %s", &count, &code, &ratio, name);

will read the data **15 p 1.575 coffee** correctly and assign the values to the variables in the order in which they appear.

Detection of errors in input

When a scanf function completes reading its list, it returns the value of number of items successfully read.

Example: **scanf("%d %f %s",&a, &b ,name);**

Will return the value 3 if the following data is typed in: **20 150.25 motor**

and return the value 1 if the following line is entered: **20 motor 150.25**

Formatted Output (printf)

We have seen the use of printf function for printing captions and numerical results.

1) Using printf we can print literal string in the form:

printf(" ");

Example: printf("hello world");

2) To print value stored at a variable, the general form is:

printf("control string",arg1,arg2,…..argn);

```
printf("%d",x );
printf("a = %f\n  b=%f ",a ,b);
printf("\n sum=%d",sum);
printf("\n\n");
```

Control string consists of three types of items:

1. Characters that will be printed on the screen as they appear.

2. Format specifications that define the output format for display of each item.
3. Escape sequence characters such as \n,\t

   A simple format specification has the following form: **%w.p type-specifier**

Where w is an integer number that specifies the total number of columns for the output value and p is another integer number that specifies the number of digits to the right of the decimal point (of a real number) or the number of character to be printed from a string. Both w and p are optional.

Output of Integer Numbers

The format specification for printing an integer number is:

%wd or %d

Where **w** specifies the minimum field width for the output.

The following examples illustrate the output of the number 9876under different formats:

Format	**output**
printf (“%d”, 9876);	9 8 7 6
printf(“%6d”, 9876);	9 8 7 6
printf(“%2d”, 9876);	9 8 7 6
printf(“%-6d”, 9876);	9 8 7 6
printf(“%06d”,9876);	0 0 9 8 7 6

It is possible to force the printing to be left –justified by placing a minus sign directly after the % character. It is also possible to pad with zero the leading blanks by placing a 0(Zero) before the field width specifier Zero (0) are known as flags.

Long integers is specified using **ld** in the place of d.

Short integers specified using **hd** in the place of d.

Output of Real Numbers

The output of real number may be displayed in decimal notation using the following format specification:

%w.p f or %f

The integer **w** indicates the minimum number of positions that are to be used for the display of the value and the integer **p** indicates the number of digits to be displayed after the decimal point(precision). The value, when displayed, is rounded to p decimal places and printed right-justified in the field of **w** columns.

We can also display a real number in exponential notation by using the specification:

%w.p e or %e

The following examples illustrate the output of the number y=98.7654 under different format specifications:

Format **output**

printf("%7.4f",y);

printf("%7.2f",y);

printf("%-7.2f",y);

printf("%f",y);

printf("%10.2e",y);

Printf("%11.4e",-y);

printf("%-10.2e",y);

printf("%e",y);

9	8	.	7	6	5	4				
		9	8	.	7	7				
9	8	.	7	7						
9	8	.	7	6	5	4	0	0		
		9	.	8	8	e	+	0	1	
-	9	.	8	7	6	5	e	+	0	1
9	.	8	8	E	+	0	1			
9	.	8	7	6	5	4	0	e	+	0	1

Printing of a single character

A single character can be displayed in a desired position using the format

%wc or %c

The character will be displayed right-justified in the field of **w** columns. We can make the display left – justified by placing a minus sign before the integer w. The default value for w is 1.

Printing of Strings

%s or %w.p s

The format specification for outputting string is similar to that of real number. It is the form **%w.ps** Where w specifies the field width for display and p instructs that only the first p characters of the string are to be displayed. The display is right –justified.

The following examples show the effect of variety of specifications in printing a string "NEW DELHI 110001", containing 16 characters (including banks)

Specification **output**

%s

N	E	W		D	E	L	H	I		1	1	0	0	0	1

%20s

				N	E	W		D	E	L	H	I		1	1	0	0	0	1

%20.10s

										N	E	W		D	E	L	H	I	

%0.5s

N	E	W		D

%-20.10s

N	E	W		D	E	L	H	I											

%5s

N	E	W		D	E	L	H	I		1	1	0	0	0	1

Mixed data output

It is permitted to mix data types in one printf statement.For example, the statement of the type

printf("%d\t%f\t%s\t%c",a,b,c,d);

Unformatted I/O functions - does not use format specifiers and therefore cannot be formatted as per our needs. Can read and write only characters and strings.

getchar(), putchar(), getch(), getche(), putch(), gets(), puts()

Reading a character

Reading a single character can be done by using the function **getchar()**.

getchar() function is a built-in function declared in stdio.h header file.

The getchar takes the following form:

```
Syntax: char_variable = getchar();
```

variable_ name is a valid C name that has been declared as **char** type.

For example:

```
char  name;
name =getchar();
```

The program shows the use of getchar function

```
#include <stdio.h>
int main()
{
        char grade;
        printf("enter grade\n");
        grade=getchar();
        printf("The grade is ");
        putchar(grade);
        return 0;
}
```

some character functions are contained in the file **ctype.h** and therefore the statement #include <ctype.h> is included in the program.

Function	Test (c is a character variable)
isalnum(c)	Is c an alphanumeric character?
isalpha(c)	Is c an alphabetic character?
isdigit(c)	Is c a digit?
islower(c)	Is c lower case letter?
isprint(c)	Is c a printable character?
ispunct(c)	Is c a punctuation mark?
isspace(c)	Is c a white space character?
isupper(c)	Is c a upper case letter?
toupper(c)	converts c to upper case
tolower(c)	converts c to lower case

WRITING A CHARACTER

putchar for writing characters one at a time to the terminal. It takes the form as shown below

```
putchar (char_variable);
```

Where **variable_name** is a type **char** containing a character. This statement displays the character contained in the variable_ name at the terminal. getchar() function is a built-in function declared in stdio.h header file.

For example:

```
answer='y';
putchar(answer);
```

will display the character 'y' on the screen.

A program that reads a character from keyboard and then prints it in reverse case that is if the input is upper case, the output will be lower case and vice versa.

The program uses three new functions: **islower**, **toupper**, and **tolower**. The function islower is a condition function and takes the value TRUE if the argument is a lowercase alphabet; otherwise takes the value FALSE. The function **toupper** converts the lowercase argument into an uppercase alphabet while the function **tolower** does the reverse.

```
#include<stdio.h>
#include<ctype.h>
int main()
{
    char alphabet;
    printf("enter an alphabet ");
    alphabet=getchar();
    if(islower(alphabet))
    {
    putchar(toupper(alphabet));
    }
    else
    {
    putchar(tolower(alphabet));
    }
    return 0;
}
```

Output

Enter an alphabet

a

A

Enter an alphabet

Q

q

Enter an alphabet

z

Z

getch(), getche(), putch() - built-in functions declared in conio.h header file.

getch() – read a single character without an echo. (i.e) the typed character is not displayed on the monitor.

Syntax: char_variable = getch();

Example: char name;
name=getch();

getche()-read a single character with an echo. (i.e) the typed character is displayed on the monitor.

Syntax: char_variable = getche();

Example: char name;
name=getche();

putch()- displays a single character on the screen.

Syntax: putch (char_variable);

Example: answer='y';
putch(answer);

Write a program to read a character and print the character using getch(), getche() and putch().

```
#include<stdio.h>
#include<conio.h>
int main()
{
    char ch1,ch2;
    printf("enter a character");
    ch1=getch();
    putch(ch1);
    return 0;
}
```

```
#include<stdio.h>
#include<conio.h>
int main()
{
    char ch1,ch2;
    printf("enter a character");
    ch1=getche();
    putch(ch1);
    return 0;
}
```

gets(), puts() - built-in functions declared in stdio.h header file.

gets() – read a string (group of characters) from standard input device like keyboard.

Syntax: gets(char_array_name);

Example:
```
char name[20];
gets(name);
```

puts() – display a string (group of characters) in the standard output device like monitor.

Syntax: puts(char_array_name);

Example:
```
char name[20];
puts(name);
```

Write a program to read your name and print the your name using gets() and puts().

```
#include<stdio.h>
int main()
{
    char name[50];
    printf("\n Enter your name");
    gets(name);
    printf("\n The name is");
    puts(name);
    return 0;
}
```