HINDUSTAN INSTITUTE OF TECHNOLOGY AND SCIENCE

SCHOOL OF COMPUTING SCIENCES

NOTES FOR

MODULE 1

PROBLEM SOLVING TECHNIQUES AND LANGUAGES

COMMON TO ALL BRANCHES

BY

Dr. Ramesh K

Professor,

School of Computing Sciences

Dr. Renjith P N

Associate Professor,

School of Computing Sciences

UNIT I
INTRODUCTION

Fundamentals of Digital Computers – Number System – Binary – Decimal – Conversion – Problems – Problem Solving Techniques: Algorithm – Pseudo code – Flow Chart.

Program Control Structures – Programming Paradigms – Generation of Programming languages – Language Translators – Features of a Good Programming Languages.

INTRODUTION:

Computer definition

- Computer is an electronic device which converts raw data into meaningful data or processed data.
- Computer is an electronic device designed in such a way that is automatically accepts and stores the input data process them and produce the desired output.
- Computer is an electronic device that operates upon information given.

Basic operations of a computer:

1. Input: It is a process of capturing the information.
2. Process: It is a transformation process to convert input into output.
3. Output: It is the result which comes from transformation process.
4. Storing: It is the process of saving the data.
5. Controlling: It is the process of directing/sequence in which all operation have to be performed.

Applications of computer:

1. Education
2. Business
3. communication
4. Science
5. Engineering
6. Entertainment
7. Banking
8. Health

Evolution of computer:

Computer is Latin word computer which means to calculate .

1. calculating machine
2. Napier bones
3. Slide rule
4. Pascal add/Sub machine.
5. Leibniz's mul/div machine.
6. Punch card system.
7. Babbage's analytical engine.

8. Hollerith’s punched card tabulating machine.

Generation of Computers:

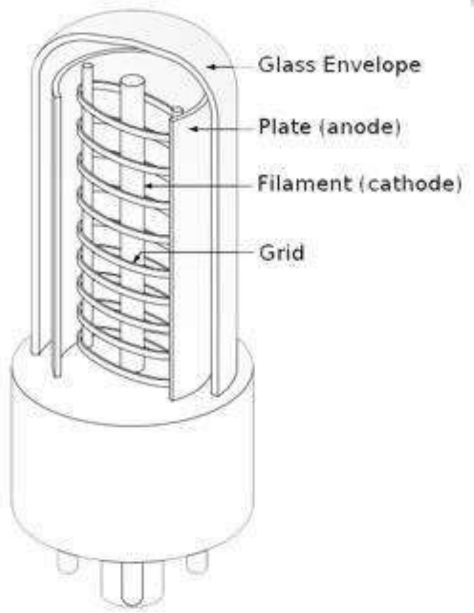
1. First generation computers
2. second generation computers
3. Third generation computers
4. Fourth generation computers
5. Fifth generation computers

Synopsis

- Hardware technology
- Software technology
- Computing characteristic
- Physical appearance
- Application
- Examples
- Advantages
- Disadvantages

First generation computers (1940-1956)

Vacuum tubes Diagram:



Hardware technology:

- They used vacuum tubes for circuitry and magnetic drums for memory.
- The Input to the computer was through punched cards.
- The output was displayed printouts.

Software technology: ❖ The Instructions were written in machine language.

- Machine language uses ’s and ’s for coding the machine instructions.
- First generation computers can solve only one problem at a time.

Computing characteristics:

- The Computation time was in milliseconds. (1 milliseconds = 1*10^{-3} Seconds).

Physical appearance:

- These computers are Enormous in size.
- They required large room for installation.

Applications:

- They were used for scientific applications.

Examples:

- UNIVAC- UNIVersal Automatic Computer
- ENIAC – Electronic numerical integrator and Computer
- EDVAC – Electronic Discrete Variable Automatic Computer

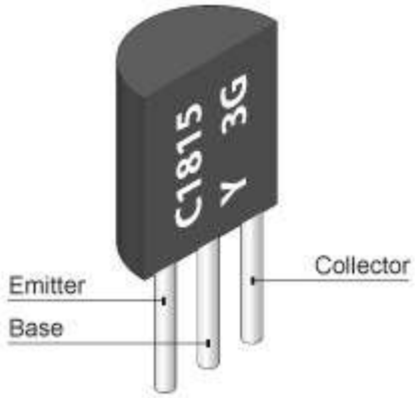
Advantages:

- They were fastest computing device in their time
- They can able to execute complex mathematical problems in efficient manner.

Disadvantages:

- First generation computers used a large number of vacuum tubes and thus generated a lot of heat.
- Because of heat these machine leads to malfunctioning.
- They consumed large amount of electricity and were expensive to operate. ❖ These computers are not reliable need high maintenance.
- It depends on machine language ’s and ’s these computers were not easy to program.
- Special purpose computers not flexible in running different type of applications.
- Single tasking not productive.
- It is not transferable because of large size.

Second generation computers (1956-1963) Transistors Diagram:



Hardware technology:

- Transistors replaced the vacuum tubes from first generation computers.
- It allows the computer become smaller faster, cheaper energy efficient and reliable.
- The second generation computer uses *magnetic core technology* for primary memory.

- They use *magnetic tapes* and *magnetic disks* for secondary memory.
- The Input to the computer was through punched cards.
- The output was displayed as printouts.
- They used the concept of stored program.
- Instructions were stored in memory of computer.

Software technology:

- Instructions were written using *assembly language.*
- Assembly language uses mnemonics like ADD, SUB, MUL, DIV, etc.
- High level programming languages such as COBOL, FORTRAN were also developed during this period.

Computing characteristics:

- The Computation time was in Micro seconds. (1 Micro seconds = 1*10^{-6} Seconds).

Physical appearance

- Transistors are smaller than vacuum tubes so size of computers is also reduced.

Applications:

- They are used for special purpose.

Examples

- PDP-8,IBM 1401,CDC 1604

Advantages:

- They are fastest computing device of their time.
- It is easier to write instructions in assembly language than machine language.
- It can be transferred easily from one location to other locations because of small size and light weight components.
- They required less maintenance than first generation computers and reliable.

Disadvantages:

- Input and output mechanisms are not improved.
- The Cost of these computers is very high beyond to use home users.
- They are special purpose computers only for specific applications.

Third generation computers (1964-1971)

Integrated Circuits Diagram:

Hardware technology:

- The third generation computers used the Integrated Circuit (IC) chips.
- In an IC Chip many transistors are place on silicon chip and it's a type of semi conductors.
- The use of IC increases the speed and efficiency of the computers.
- The Input to the computer was through keyboard.
- The output was displayed in monitors.

Software technology:

- The Keyboard monitors are interfaced through operating system.
- Operating system allowed multiple applications run at same time.
- High level languages were used extensively for programming instead of machine languages and assembly languages.

Computing characteristics:

- The Computation time was in nanoseconds. (1 nanoseconds = $1*10^{-9}$ Seconds).

Physical appearance:

- Size of the computers was quit small compared to the second generation computers.

Application:

- These Computers become accessible to mass audience.
- They are produced commercially.
- They are general purpose computers and it can be used for any purpose.

Example:

- IBM System 370, PDP 11

Advantages:

- The third generation computers used less power and generated less heat than the second generation computers.
- The cost of the computers reduced significantly as individual components of the computer were not required to assemble manually.
- The maintenance cost of the computer was also less compared to previous generation computers.
- They were fastest computing device compared to first and second generation computers.
- It can be transferred easily from one location to other locations because of small size.
- These computers are productive because of less computing time.
- It is easier to program in high level language than any other language.
- These computers are multi tasking and general purpose computers.

Disadvantages

- The storage capacity of the computer is small.
- The performance of the computer got degraded while executing large applications and complex calculations.

- These computers were needed to be placed in Air-conditioned room.

High level language:

- It is a computer programming language that is independent of machine details.
- The Programmers find it easy to use them.
- The Programs written in one computer can be executed in other computers.

Fourth generation computers (1971 to present)- Micro processors

Hardware technology

- They use the large scale integration and the very large scale integration technology.
- Thousands of transistors are integrated on a small silicon chip using LSI technology.
- VLSI allows hundreds of transistors are integrated

Software technology

- Several operating systems like the MS-DOS and MS –windows developed during this time.
- This generation of computer supports Graphical user interface (GUI).
- GUI is a user friendly interface that allows user to interact with the computer via menus and icons,
- High level languages are used for writing of programs.

Computing characteristic

- The computation time is in Pico seconds (1 Pico seconds = $1*10^{-12}$ Seconds).

Physical appearance

- They are smaller than the computers of the previous generation. Some can even fit into the palm of the hands

Application

- They become widely available for commercial purpose
- Personal computers become available to the home users.

Examples

- The Intel 14004 chip was the first microprocessor.
- The components of the computers like CPU and memory were located in a single chip.

Advantages

- The micro processor has resulted in the fourth generation computers being smaller and cheaper than their predecessors.
- Storage capacity is very large and faster.

- They are portable and more reliable.
- They generate much lesser heat and require less maintenance compared to their predecessors.
- Networking has resulted in resource sharing and communication among different computers.
- It provides user friendly environment because of GUI and input /output devices.

Disadvantages

- Soldering of LSI and VLSI chips on the board was not easy process and it needs a complex technology.
- Working of these computers depends on instructions given by programmers. **Fifth generation computers (Present- Next generation)**

Hardware technology

- The goal of fifth generation computer is to develop computers that are capable of learning and self –organization.
- They use super large scale integrated (SLSI) chips that are able to store millions of components on one silicon chip.
- They have large memory requirements.

Software technology

- They use parallel processing that allows several instructions to be executed in parallel, instead of serial execution.
- Parallel processing results in faster processing speed.
- The Intel dual core micro processor uses parallel processing.
- They are based on Artificial intelligence; they try to simulate human way of thinking and reasoning.
- AI includes areas like expert system, natural language processing (NLP), Speech recognition, voice recognition, and robotics.

Computing characteristic

- Computing time is less than Pico seconds.

Physical appearance

- It is available according to the user needs. Laptop, PC, PDA

Application

- General purpose.

Examples

- Apple, MAC, Notepad, Tab.

Advantages

- Fastest computing device.
- Many applications can be run on same time (high speed).
- VLSI technology leads to reduced size, these computers can be used in travelling.
- Multimedia features and wireless are available.

Disadvantages

- Not provided intelligence program that can guide computers in different applications.

Classifications of computers

- Based on their size and type
- Based On Operating Principles
- Based on area of applications
- Based on number of users

Based on their size and type:

1. Micro computers
2. Mini computers
3. Mainframe computers
4. Super computers

Microcomputers (Personal Computer)

A microcomputer is the smallest general purpose processing system. The older pc started 8 bit processor with speed of 3.7MB and current pc 64 bit processor with speed of 4.66 GB.

Examples: - **IBM PC**s, **APPLE** computers

Microcomputer can be classified into 2 types:

1. Desktops
2. Portables

The difference is portables can be used while travelling whereas desktops computers cannot be carried around.

The different portable computers are: - 1.
Laptop

2. Notebooks
3. Palmtop (hand held)
4. Wearable computers **Laptop**: -

This computer is similar to a desktop computers but the size is smaller. They are expensive than desktop. The weight of laptop is around 3 to 5 kg.

Notebook: -

These computers are as powerful as desktop but size of these computers are comparatively smaller than laptop and desktop. They weigh 2 to 3 kg. They are more costly than laptop.

Palmtop (Hand held): -

They are also called as personal Digital Assistant (PDA). These computers are small in size. They can be held in hands. It is capable of doing word processing, spreadsheets and hand writing recognition, game playing, faxing and paging. These computers are not as powerful as desktop computers.

Ex: - 3com palmV.

Wearable computer: -

The size of this computer is very small so that it can be worn on the body. It has smaller processing power. It is used in the field of medicine. For example pace maker to correct the heart beats. Insulin meter to find the levels of insulin in the blood.

Minicomputer:

A minicomputer is a medium-sized computer. That is more powerful than a microcomputer. These computers are usually designed to serve multiple users simultaneously (Parallel Processing). They are more expensive than microcomputers. Examples: Digital Alpha, Sun Ultra.

Mainframe computers:

Computers with large storage capacities and very high speed of processing (compared to mini- or microcomputers) are known as mainframe computers. They support a large number of terminals for simultaneous use by a number of users like ATM transactions. They are also used as central host computers in distributed data processing system.

Examples: - IBM 370, S/390.

Supercomputer:

Supercomputers have extremely large storage capacity and computing speeds which are many times faster than other computers. A supercomputer is measured in terms of tens of millions Instructions per second (mips), an operation is made up of numerous instructions. The supercomputer is mainly used for large scale numerical problems in scientific and engineering disciplines such as Weather analysis. Examples: - IBM Deep Blue

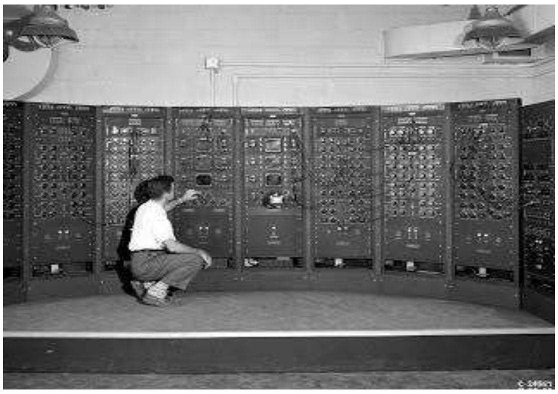
Classification Based On Operating Principles

1. Digital Computers
2. Analog Computers
3. Hybrid Computers **Digital Computers: -**

Operate essentially by counting. All quantities are expressed as discrete or numbers. Digital computers are useful for evaluating arithmetic expressions and manipulations of data (such as preparation of bills, ledgers, solution of simultaneous equations etc)

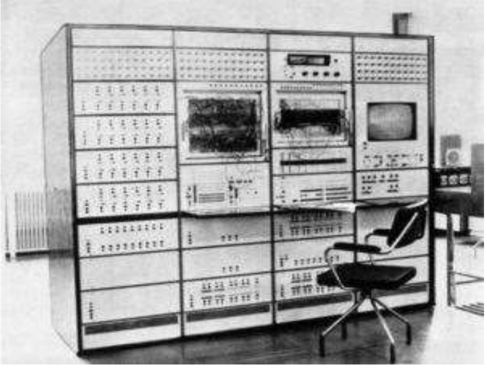
Analog Computers: -

An analog computer is a form of computer that uses the continuously changeable aspects of physical phenomena such as electrical, mechanical, or hydraulic quantities to model the problem being solved. In contrast, digital computers represent varying quantities symbolically, as their numerical values change.



Hybrid Computers: -

Hybrid Computers are computers that exhibit features of analog computers and digital computers. The digital component normally serves as the controller and provides logical operations, while the analog component normally serves as a solver of differential equations.



Classification based on area of applications

Modern computers depending upon their applications are classified as: -

1. Special Purpose Computers
2. General Purpose Computers

Special Purpose Computers:

A special purpose computer is designed only to meet the requirements of a particular task or application. The instructions needed to perform a particular task are permanently stored into the internal memory, so that it can perform the given task on a single command. It therefore doesn't possess unnecessary options and is less expensive.

General Purpose Computers: -

General Purpose computers are designed to meet the needs of many different applications. In these computers, the instructions needed to perform a particular task are wired permanently into the internal memory. When one job is over, instructions for another job can be loaded into the internal memory for processing. This, a general purpose machine can be used to prepare pay-bills, manage inventories, print sales report and so on.

Classification based on word length and number of users

1. Single user
2. Multi user
3. Network

Single User: - Only one user can use the resource at any time.

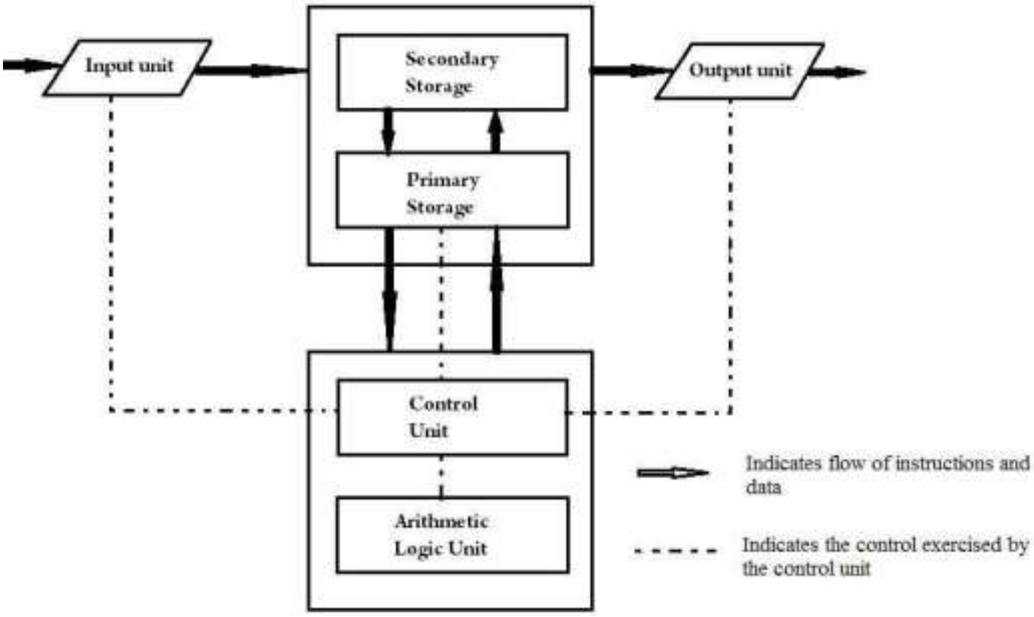
Multi User: -
A single computer shared by a number of users at any time.

Network: -

A number of interconnected autonomous computers shared by a number of users at any time.

BASIC ORGANIZATION OF COMPUTERS:

- This topic explains the way in which different units of computer are interconnected with each other and controlled.
- It describes the various operations performed by each units.


Input unit
Secondary Storage
Output unit
Primary Storage
Control Unit
Arithmetic Logic Unit
Indicates flow of instructions and data
Indicates the control exercised by the control unit

It has four units

1. Input unit
2. Output unit
3. CPU
4. Memory unit

- ❖ This unit contains devices with the help of which we enter data into computer. This unit makes link between user and computer.
- ❖ The input devices translate the information into the form understandable by computer.

1. Keyboard
2. Mouse
3. Joy Stick
4. Light pen
5. Track Ball
6. Scanner
7. Graphic Tablet
8. Microphone
9. Webcam
10. Bar Code Reader

Key Board

- ❖ Keyboard is the most common and very popular input device which helps in inputting data to the computer.
- ❖ The layout of the keyboard is like that of traditional typewriter, although there are some additional keys provided for performing additional functions.
- ❖ Keyboards are of two sizes 84 keys or 101/102 keys, but now keyboards with 104 keys or 108 keys are also available for Windows and Internet.

The keys on the keyboard are as follows:

Keys	Description
Typing Keys	These keys include the letter keys (A-Z) and digit keys (0-9) which generally give same layout as that of typewriters.
Numeric Keypad	It is used to enter numeric data. Generally, it consists of a set of 17 keys that are laid out in the same configuration used by most adding machines and calculators.
Function Keys	The twelve function keys are present on the keyboard which is arranged in a row at the top of the keyboard. Each function key has unique meaning and is used for some specific purpose.
Control keys	These keys provide cursor and screen control. It includes four directional arrow keys. Control keys also include Home, End, Insert, Delete, Page Up, Page Down, Control(Ctrl), Alternate(Alt), Escape(Esc).
Special Purpose Keys	Keyboard also contains some special purpose keys such as Enter, Shift, Caps Lock, Num Lock, Space bar, Tab, and Print Screen.

Mouse

- Mouse is most popular pointing device. It is a very famous cursor-control device having a small palm size box with a round ball at its base which senses the movement of mouse and sends corresponding signals to CPU when the mouse buttons are pressed.
- Generally it has two buttons called left and right button and a wheel is present between the buttons. Mouse can be used to control the position of cursor on screen, but it cannot be used to enter text into the computer.

Advantages

- Easy to use
- Not very expensive
- Moves the cursor faster than the arrow keys of keyboard.

Scanner

- Scanner is an input device which works more like a photocopy machine.
- It is used when some information is available on a paper and it is to be transferred to the hard disc of the computer for further manipulation.

- Scanner captures images from the source which are then converted into the digital form that can be stored on the disc. These images can be edited before they are printed.
- It converts hard copy into soft copy.

Output unit:

Following are few of the important output devices which are used in a computer

- Monitors
- Graphic Plotter
- Printer

Monitor

Monitors, commonly called as Visual Display Unit (VDU), are the main output device of a computer. It forms images from tiny dots, called pixels that are arranged in a rectangular form. The sharpness of the image depends upon the number of pixels. There are two kinds of viewing screen used for monitors.

- Cathode-Ray Tube (CRT)
- Flat- Panel Display (LED, LCD)

Printers:

It is one of the output device which convert text which displays on the screen into paper sheets, which means it converts soft copy into hard copy.

Types of printers:

1. Dot matrix printers
2. Ink jet printers
3. Laser printers
4. Poster printers

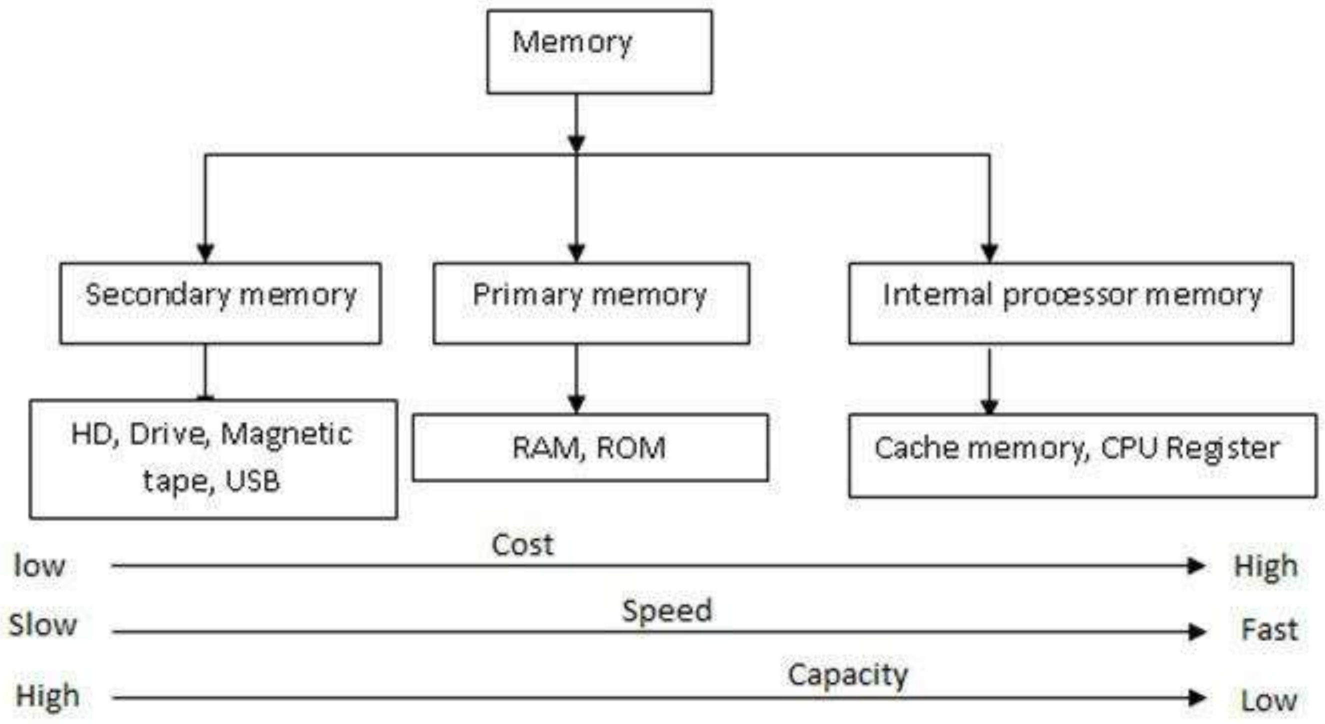
Speakers:

- It is a kind of output device which convert electrical signal into sound.
- It may be inbuilt or externally attached.
- The sound card and audio driver should be attached to the computers to produce audio output.

MEMORY UNIT

Memory is primarily of three types

- Primary Memory/Main Memory
- Secondary Memory
- Internal processing memory/Cache Memory



The memory unit of the computer is used to store the following,

- Data
- Instructions required for processing the data
- Intermediate results of processing
- Final processed results

Primary Memory

Primary memory holds only those data and instructions on which computer is currently working. It has limited capacity and data is lost when power is switched off. It is generally made up of semiconductor device. These memories are not as fast as registers. The data and instruction required to be processed reside in main memory. It is divided into two subcategories RAM and ROM.

RAM(Random Access Memory)	ROM(Read-only memory)
RAM is a form of data storage that can be accessed randomly at any time	ROM is also a form of data storage that cannot be easily altered or reprogrammed.
It allows reading and writing.	It only allows reading.
RAM is volatile i.e. its contents are lost when the device is powered off.	It is non-volatile i.e. its contents are retained even when the device is powered off.
It is used as main memory of a computer.	It is used to store BIOS (Basic input output system) of a computer.

Characteristics of Main Memory

- These are semiconductor memories.
- Usually volatile memory.
- Data is lost in case power is switched off.
- It is working memory of the computer.

- Faster than secondary memories.
- A computer cannot run without primary memory.

Secondary Memory

This type of memory is also known as external memory or non-volatile. It is slower than main memory. These are used for storing data/Information permanently. CPU directly does not access these memories instead they are accessed via input-output routines. Contents of secondary memories are first transferred to main memory, and then CPU can access it. For example: disk, CD-ROM, DVD etc.

Characteristic of Secondary Memory

- These are magnetic and optical memories.
- It is known as backup memory.
- It is non-volatile memory.
- Data is permanently stored even if power is switched off.
- It is used for storage of data in a computer.
- Computer may run without secondary memory.
- Slower than primary memories.

Cache Memory

Cache memory is a very high speed semiconductor memory which can speed up CPU. It acts as a buffer between the CPU and main memory. It is used to hold those parts of data and program which are most frequently used by CPU. The parts of data and programs are transferred from disk to cache memory by operating system, from where CPU can access them.

The advantages of cache memory are as follows

- Cache memory is faster than main memory.
- It consumes less access time as compared to main memory.
- It stores the program that can be executed within a short period of time.
- It stores data for temporary use.

The disadvantages of cache memory are as follows

- Cache memory has limited capacity.
- It is very expensive.

CPU (CENTRAL PROCESSING UNIT)

CPU is considered as the brain of the computer. CPU performs all types of data processing operations. It stores data, intermediate results and instructions (program).It controls the operation of all parts of computer.

CPU performing following four main operations:

- Fetching instructions from memory.
- Decoding the instructions to decide what operations are to be performed.
- Executing the instructions.
- Storing the results back to the memory.

CPU itself has following three components

1. ALU(Arithmetic Logic Unit)
2. CU (Control Unit)
3. Registers

ALU (Arithmetic Logic Unit)

This unit consists of two subsections namely

- Arithmetic section
- Logic Section

Arithmetic Section

Function of arithmetic section is to perform arithmetic operations like addition, subtraction, multiplication and division. All complex operations are done by making repetitive use of above operations.

Logic section

- Function of logic section is to perform logic operations such as comparing, selecting, matching and merging of data.
- It will perform 6 operations like AND,OR,NOT, >,<… etc

Control Unit:

- It maintains the sequence of operations being performed by CPU.
- It guides ALU about which operations to be performed.
- It uses two registers namely Program counter registers and status registers.

Registers:

- Register is a small storage area inside the CPU from where the data is retrieved faster than any other storage device.
- They are high speed memory locations used for holding data, instructions, and intermediate results.

Types of registers:

1. PC – Program counter
2. IR – Instructions registers
3. MAR – Memory Address registers
4. MBR – Memory Buffer registers
5. MDR – Memory Data registers
6. ACC - Accumulator

NUMBER SYSTEM:

Number system will be having following two values.

1. Base value
2. Position value

   (Base $^{\text{position}}$)

Types of number system:

1. Decimal number system
2. Binary number system
3. Octal number system
4. Hexadecimal number system

Decimal number system (10^{position})

- It consists of 10 digits 0-9.
- All number should be represented as a combination of 0-9. Ex: 5895

Position will differ as following

3	2	1	0	.	-1	-2	-3
10_3	10_2	10_1	10_0		10_{-1}	10_{-2}	10_{-3}

Binary number system (2^{position})

- It consists of 10 digits 0-1.
- All number should be represented as a combination of 0 and 1. Ex: 01110

Position will differ as following

3	2	1	0	.	-1	-2	-3
2_3	2_2	2_1	2_0		2_{-1}	2_{-2}	2_{-3}

Octal number system (8^{position})

- It consists of 10 digits 0-7.
- All number should be represented as a combination of 0-7. Ex: 70564

Position will differ as following

3	2	1	0	.	-1	-2	-3
8_3	8_2	8_1	8_0		8_{-1}	8_{-2}	8_{-3}

Hexadecimal number system (16^{position})

- It consists of 10 digits 0-15.
- 0-9 and A, B, C, D, E, F
- All number should be represented as a combination of these 16 digits. Ex: 3AB

Position will differ as following

3	2	1	0	.	-1	-2	-3
16_3	16_2	16_1	16_0		16_{-1}	16_{-2}	16_{-3}

Summary:

Number system	Base	Digit	Largest digit
Decimal	10	0-9	9
Binary	2	0,1	1
Octal	8	0-7	7
Hexadecimal	16	0-9, A, B, C, D, E, F	F

Conversion table:

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

Types of Conversion:

1. Decimal to Binary
2. Decimal to Octal
3. Decimal to Hexadecimal
4. Binary to Decimal
5. Octal to Decimal
6. Hexadecimal to Decimal
7. Binary to Octal
8. Binary to Hexadecimal
9. Octal to Binary
10. Hexadecimal to Binary
11. Octal to Hexadecimal
12. Hexadecimal to Octal

Decimal to Binary

1. 147	= 10010011
2. 94	=1011110
3. 675	=1010100011
4. 94.2345	=1011110.0011

5. 150.64 =10010110.1010
6. 24.14 =11000.0010
7. 4064.865 =100000110000.11011 **Decimal to**

Octal:

1. 147 =223
2. 675 =1243
3. 22.33 =26.2507
4. 4064.865 =7740.6727 **Decimal to Hexadecimal:**

1. 675 =2A3
2. 4064.865 =F40.DD70A

Binary to Decimal

1. 1011 =11
2. 1010100010=675
3. 10010110 =150
4. 1011.1101 =11.8125
5. 11000.0010=24.125

Octal to Decimal

1. 62 =50
2. 24.36 =20.46875
3. 226.5075 =150.6398
4. 30.1075 =24.1398
5. 26.250 =22.3281

Hexadecimal to Decimal

1. 2A3 =675
2. 96.A70A =150.6525

Binary to Octal:

1. 1110101100110 =16546
2. 100111011 =473
3. 1010111010011 =12723 **Binary to Hexadecimal:**
1. 1110101100110 =1D66
2. 100111011 =13B
3. 1010111010011 =15D3 **Octal to Binary:**

1. 16546 =1110101100110
2. 6314 =110011001100 **Hexadecimal to Binary:**
1. 1D66 =1110101100110
2. 2A3 =001010100011
3. FAE1 =1111101011100001

Octal to Hexadecimal:

1. 365	=F5
2. 1243	=2A3

Hexadecimal to Octal:

1. F5	=365
2. 18	=30

Gray code to Binary (add with results):

1. 11010011	=10011101

Binary to gray code (add with previous value):

1. 10011101	=11010011

PROBLEM SOLVING TECHNIQUES:

1. Algorithm
2. Flow chart
3. pseudo code

Algorithm

- The solution to any computing problem involves executing a series of actions in a specific order.
- A procedure for solving a problem in terms of the actions to be executed, and the order in which these actions are to be executed is called an algorithm.
- Correctly specifying the order in which the actions are to be executed is important.
- Algorithm is an ordered sequence of finite, well-defined, Unambiguous instructions for completing a task.
- It is a step by step procedure for solving a task or problem.
- Algorithm helps a programmer in breaking down the solution of a problem into a no of sequential steps.
- Corresponding to each step a statement is written in a programming language.

Characteristics of algorithm:

- Each and every instruction should be simple.
- Instructions should not repeat infinity.
- Algorithm should ultimately terminate.
- It should be in sequence.
- It should be in normal English.
- Derived results should be obtained only after the algorithm terminates.

Example:

Add two numbers:

Step 1: Start

Step 2: Read values num1 and num2.

Step 3: Add num1 and num2 and assign the result to sum.

sum←num +num

Step 4: Display sum

Step 5: Stop

Greatest of three numbers:

Step1: Start

Step2: Read three numbers A, B, C

Step3: Compare A with B

Step4: If A is larger compare it with C

Step5: If A is larger than C then A is the largest otherwise C is the largest.

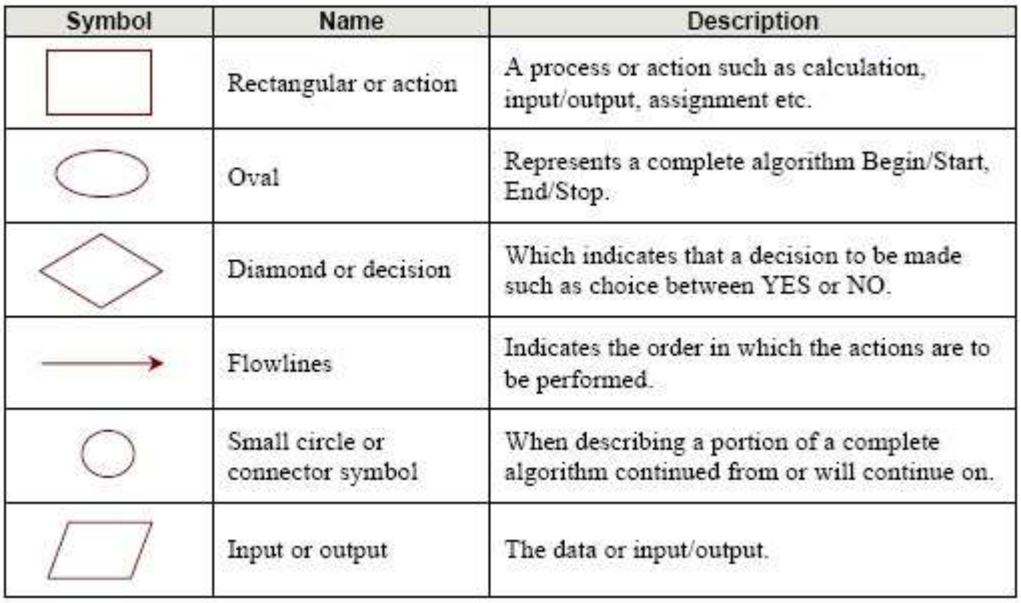
Step6: If A is smaller than or equal to B in the first step then B is compared with C. Step7: If B is larger than C then B is the largest number otherwise C is the largest number.

Step8: Stop

Flow chart

Flow chart is defined as graphical representation of the logic for problem solving

The purpose of flowchart is make the logic of the program clear in a visual form



Symbol	Name	Description
▭	Rectangular or action	A process or action such as calculation, input/output, assignment etc.
⬭	Oval	Represents a complete algorithm Begin/Start, End/Stop.
◇	Diamond or decision	Which indicates that a decision to be made such as choice between YES or NO.
→	Flowlines	Indicates the order in which the actions are to be performed.
○	Small circle or connector symbol	When describing a portion of a complete algorithm continued from or will continue on.
▱	Input or output	The data or input/output.

Table 6.1: Some of the flowchart symbols used in this Module.

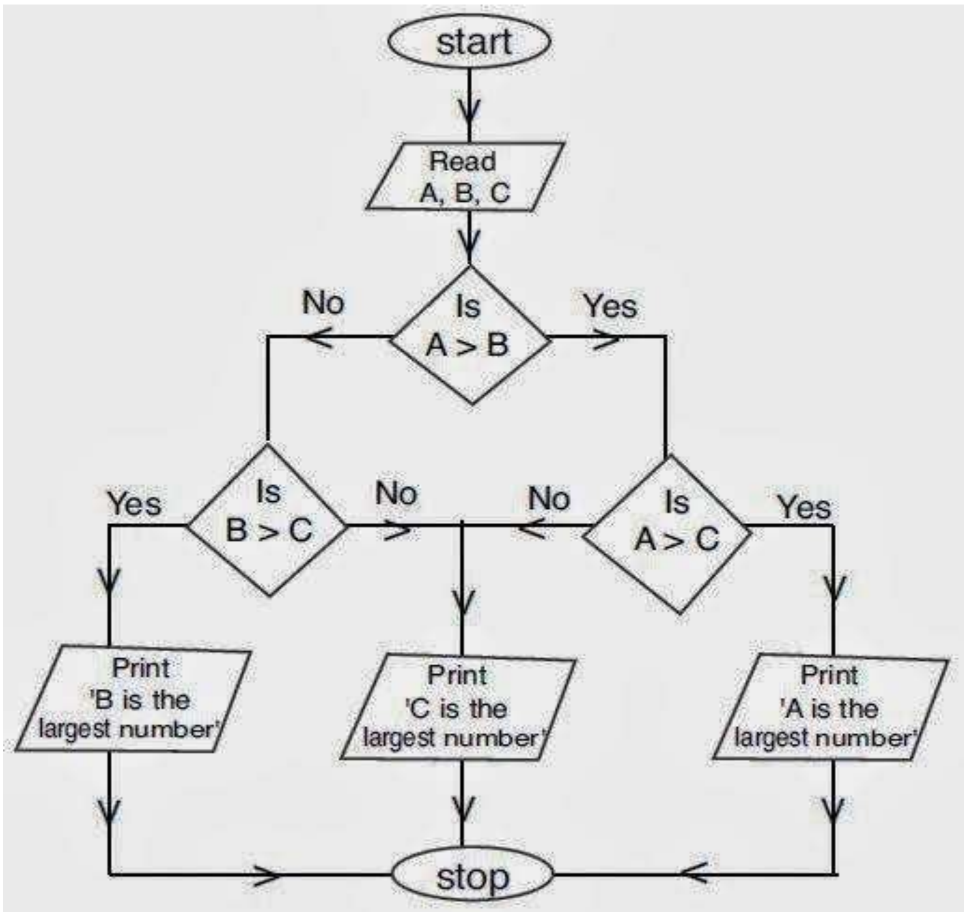
Advantages of flowchart:

1. **Communication: -** Flowcharts are better way of communicating the logic of a system to all concerned.
2. **Effective analysis: -** With the help of flowchart, problem can be analyzed in more effective way.

3. **Proper documentation: -** Program flowcharts serve as a good program documentation, which is needed for various purposes.
4. **Efficient Coding: -** The flowcharts act as a guide or blueprint during the systems analysis and program development phase.
5. **Proper Debugging: -** The flowchart helps in debugging process.
6. **Efficient Program Maintenance: -** The maintenance of operating program becomes easy with the help of flowchart. It helps the programmer to put efforts more efficiently on that part.

Disadvantages of flow chart:

1. **Complex logic: -** Sometimes, the program logic is quite complicated. In that case, flowchart becomes complex and clumsy.
2. **Alterations and Modifications: -** If alterations are required the flowchart may require re-drawing completely.
3. **Reproduction: -** As the flowchart symbols cannot be typed, reproduction of flowchart becomes a problem.
4. The essentials of what is done can easily be lost in the technical details of how it is done.



Pseudo code:

- Pseudocode is an artificial and informal language that helps you develop algorithms.
- Pseudocode is similar to everyday English; it's convenient and user friendly although it's not an actual computer programming language.
- Pseudocode programs are not executed on computers.
- It gives us the sketch of the program before actual coding.
- There is no standard syntax for pseudo code.

Preparing pseudo code:

- Standards used for representing various action are,
- Input: INPUT, GET, READ
- Output: OUTPUT, PRINT, DISPLAY
- Calculation: COMPUTE, CALCULATE, ADD, SUB, MUL..

Advantages:

- It can be easily in any word processor.
- It can be easily modified as compared to flowchart.
- Its implementation is very useful in structured design elements.
- It can be written easily.
- It can be read and understood easily.
- Converting a pseudo code to programming language is very easy as compared with converting a flowchart to programming language.

Disadvantages:

- It is not visual.
- We do not get a picture of the design.
- There is no standardized style or format, so one pseudo code may be different from another.
- For a beginner, It is more difficult to follow the logic or write pseudo code as compared to flowchart.

Example:

```
BEGIN
INPUT num 1,num 2
SUM = num 1 + num 2
DISPLAY total
END
```

Algorithm	**Flowchart**	**Pseudo code**
An algorithm is a sequence of instructions used to solve a problem	It is a graphical representation of algorithm	It is a language representation of algorithm.
User needs knowledge to write algorithm.	not need knowledge of program to draw or understand flowchart	Not need knowledge of program language to understand or write a pseudo code.

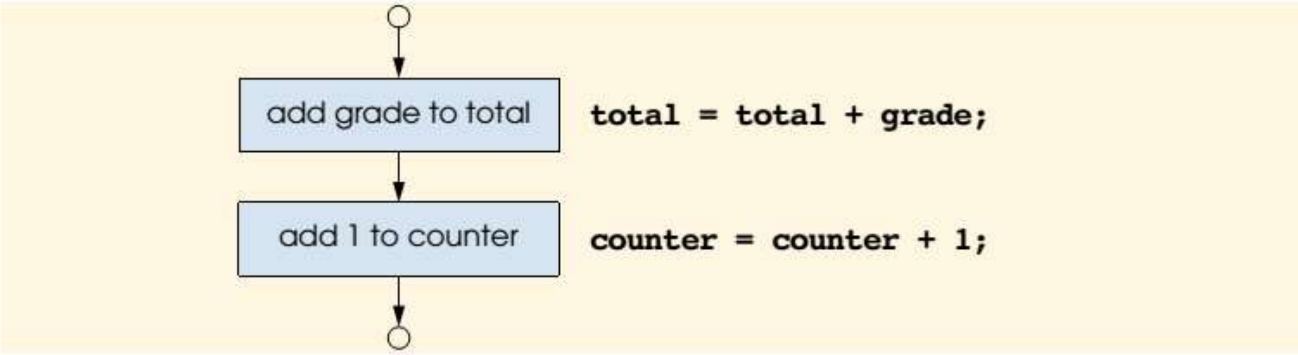
Program Control Structures:

Flow of control through any given function is implemented with three basic types of control structures:

1. Sequential Control Structure
2. Selection Control Structure
3. Repetition Control Structure

1. Sequential:

This is the default logic used by every compiler. In this logic the program instructions are executed in the order in which they appeared in the program. The compiler scans the program, instruction by instruction and run the instruction one by one from top to bottom. This is also called linear logic. Sequential execution of code statements (one line after another) -- like following a recipe.



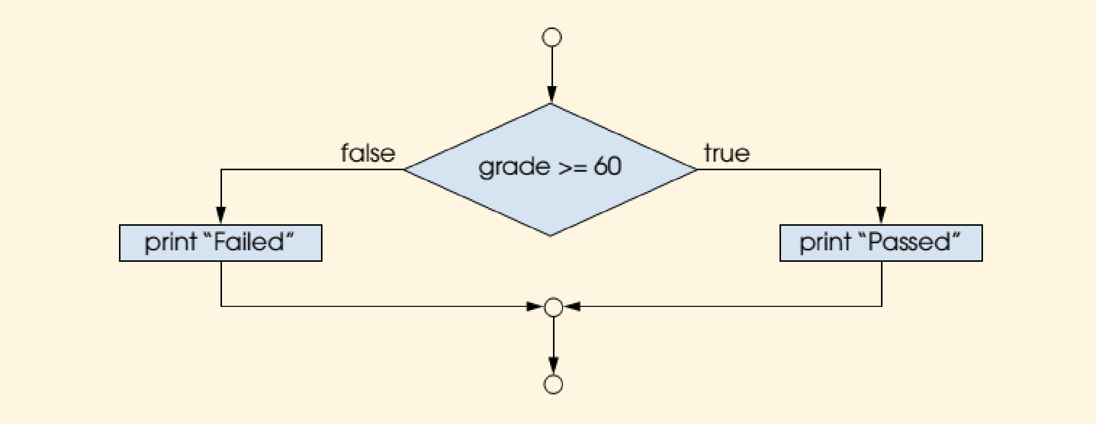
2. **Selection (Decision Making):**

Selection used for decisions, branching -- choosing between 2 or more alternative paths. In C, these are the types of selection statements:

if

if/else

switch



3. **Repetition (Loops):**

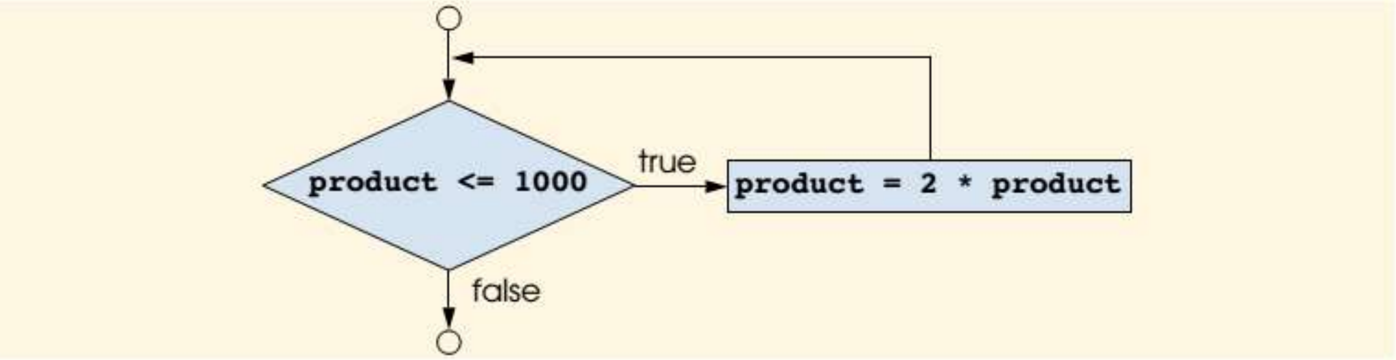
Repetitions are used for looping, i.e. repeating a piece of code multiple times in a row. In C, there are three types of loops:

while

do/while

for

The function construct, itself, forms another way to affect flow of control through a whole program.



*Detailed explanation of Program Controls with examples will be explained in Unit II.

Programming Paradigms:

A programming paradigm is a way or style of how computer programs are written. There are three fundamentally different programming paradigms:

- Procedural Programming Paradigm
- Object-Oriented Programming Paradigm
- Flow Programming

1. **The procedural programming paradigm**

The procedural programming paradigm (also called imperative programming) is concerned with defining a linear procedure or sequence of programming statements. A key feature of the paradigm is the partitioning of functionality into small discrete re-usable modules called procedures (subroutines or functions) that act like small programs themselves with their own scope, inputs and outputs. A procedural code example is executed from a single point of control or entry point which calls out into declared procedures, which in turn may call other procedures.

Procedural programming was an early so-called 'high-level programming paradigm' (compared to lower-level machine code) and is the most common and well understood form of programming. Newer paradigms (such as Object-Oriented programming) and modern businesses programming languages (such as C++, Java and C#) are built on the principles of procedural programming.

All algorithms in this book were implemented using a procedural programming paradigm in the Ruby Programming Language. A procedural representation was chosen to provide the most transferrable instantiation of the algorithm implementations. Many languages support the procedural paradigm and procedural code examples are expected to be easily ported to popular paradigms such as object-oriented and functional.

2. Object-oriented Programming

The Object-Oriented Programming (OOP) paradigm is concerned with modeling problems in terms of entities called objects that have attributes and behaviors (data and methods) and interact with other entities using message passing (calling methods on other entities). An object developer defines a class or template for the entity, which is instantiated or constructed and then may be used in the program.

Objects can extend other objects, inheriting some or all of the attributes and behaviors from the parent providing specific modular reuse. Objects can be treated as a parent type (an object in its inheritance tree) allowing the use or application of the objects in the program without the caller knowing the specifics of the behavior or data inside the object. This general property is called polymorphism, which exploits the encapsulation of attributes and behavior within objects and their capability of being treated (viewed or interacted with) as a parent type.

Organizing functionality into objects allows for additional constructs such as abstract types where functionality is only partially defined and must be completed by descendant objects, overriding where descending objects re-define behavior defined in a parent object, and static classes and behaviors where behavior is executed on the object template rather than the object instance.

There are common ways of solving discrete problems using object-oriented programs called patterns. They are organizations of behavior and data that have been abstracted and presented as a solution or idiom for a class of problem. The Strategy Pattern is an object-oriented pattern that is suited to implementing an algorithm. This pattern is intended to encapsulate the behavior of an algorithm as a strategy object where different strategies can be used interchangeably on a given context or problem domain. This strategy can be useful in situations where the performance or capability of a range of different techniques needs to be assessed on a given problem (such as algorithm racing or bake-offs). Additionally, the problem or context can also be modeled as an interchangeable object, allowing both algorithms and problems to be used interchangeably. This method is used in object-oriented algorithm frameworks.

3. **Flow Programming**

Flow, data-flow, or pipeline programming involves chaining a sequence of smaller processes together and allowing a flow of information through the sequence in order to perform the desired computation. Units in the flow are considered black-boxes that communicate with each other using message passing. The information that is passed between the units is considered a stream and a given application may have one or more streams of potentially varying direction. Discrete information in a stream is partitioned into information packets which are passed from unit-to-unit via message buffers, queues or similar data structures.

A flow organization allows computing units to be interchanged readily. It also allows for variations of the pipeline to be considered with minor reconfiguration. A flow or pipelining structure is commonly used by software frameworks for the organization within a given algorithm implementation, allowing the specification of operators that manipulate candidate solutions to be varied and interchanged.

Generations of Programming Language

- The first generation languages, or 1GL, are low level languages that are machine language.

- The second generation languages, or 2GL, are also low-level languages that generally consist of assembly languages.

- The third generation languages, or 3GL, are high level languages such as C.

The fourth generation languages, or 4GL, are languages that consist of statements similar to statements in a human language. Fourth generation languages are commonly used in database programming and scripts.

- The fifth generation languages, or 5GL, are programming languages that contain visual tools to help develop a program. A good example of a fifth generation language is Visual Basic.

Types of Programming Language

There are three types of programming language:

1. Machine language (Low-level language)

2. Assembly language (Low-level language)

3. High-level language

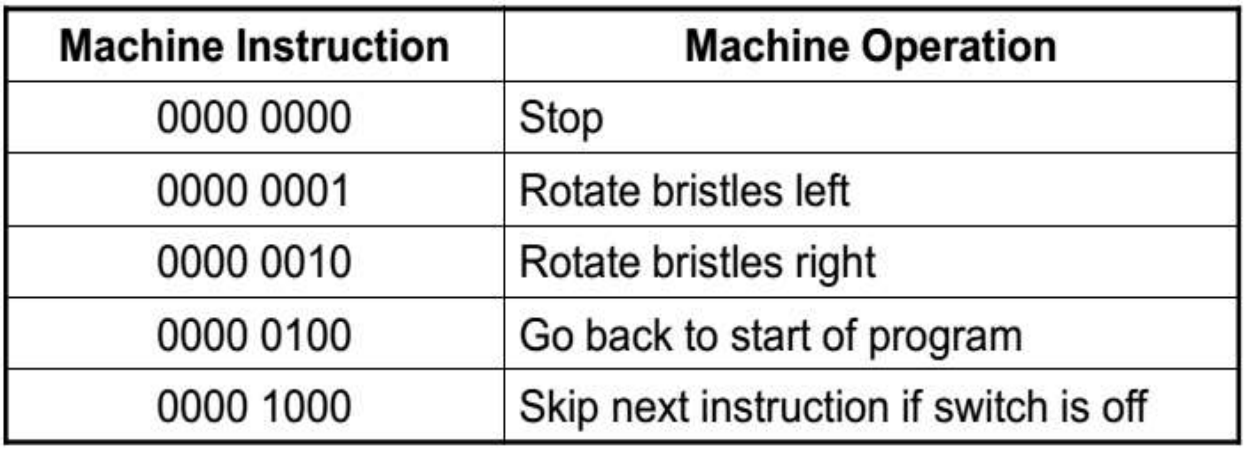
1. Machine Language:

Low-level languages are closer to the language used by a computer, while high-level languages are closer to human languages.

• Machine language is a collection of binary digits or bits that the computer reads and interprets.

• Machine languages are the only languages understood by computers.

• While easily understood by computers, machine languages are almost impossible for humans to use because they consist entirely of numbers.



Machine Instruction	Machine Operation
0000 0000	Stop
0000 0001	Rotate bristles left
0000 0010	Rotate bristles right
0000 0100	Go back to start of program
0000 1000	Skip next instruction if switch is off

2. Assembly language (Low-level language)

A program written in assembly language consists of a series of instructions mnemonics that correspond to a stream of executable instructions, when translated by an assembler, that can be loaded into memory and executed.

Assembly languages use keywords and symbols, much like English, to form a programming language but at the same time introduce a new problem. The problem is that the computer doesn't understand the assembly code, so we need a way to convert it to machine code, which the computer does understand. Assembly language programs are translated into machine language by a program called an assembler.

Example:

Machine language : 10110000 01100001

Assembly language : mov a1, #061h

Meaning:

Move the hexadecimal value 61 (97 decimal) into the processor register named "a1".

High Level Language

High-level languages allow us to write computer code using instructions resembling everyday spoken language (for example: print, if, while) which are then translated into machine language to be executed.

Programs written in a high-level language need to be translated into machine language before they can be executed. Some programming languages use a compiler to perform this translation and others use an interpreter.

Examples of High-level Language:

- ADA
- C
- C++
- JAVA
- BASIC
- COBOL
- PASCAL
- PHYTON

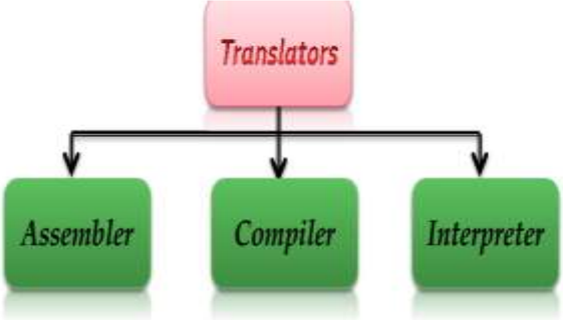
Language Translator

Language translators convert programming source code into language that the computer processor understands.

Programming source code has various structures and commands, but the computer processors understand only machine language.

Language Translator are of three types:

1. Assembler
2. Compiler
3. Interpreter



1. COMPILER

A computer program that translates source code into object code.

Source code: - High-level language version of the program.

Object code: -The resulting machine code program.

Primary reason for compiling source code is to create an executable program.

It checks all kinds of limits, ranges, errors etc. before executing it completely but the disadvantage is that when an error in a program occurs it is difficult to pin-point its source in the original program.

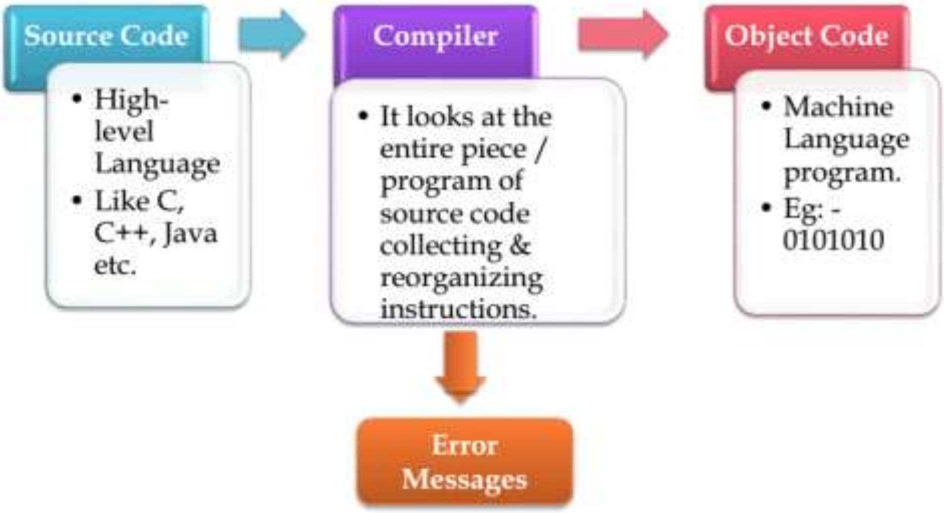

Source Code
• High-level Language
• Like C, C++, Java etc.
Compiler
• It looks at the entire piece / program of source code collecting & reorganizing instructions.
Object Code
• Machine Language program.
• Eg: - 0101010
Error Messages

Fig. COMPILER

2. *Interpreter:*

An interpreter is closely related to a compiler, but takes both source program and input data. The basic purpose of interpreter is same as that of complier but it can't create a executable file like compiler.

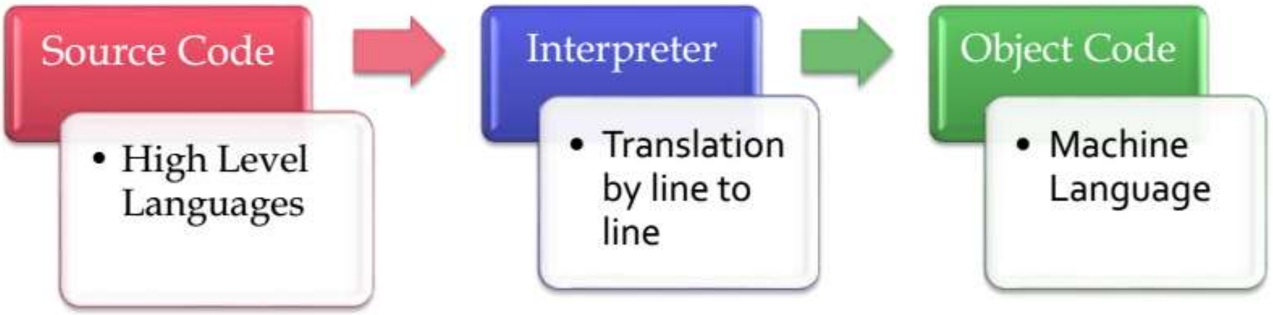


Fig. Interpreter

3. Assembler

Assembler is software or a tool that translates Assembly language to machine code.

▪ Assembly is a human readable language but it typically has a one to one relationship with the corresponding machine code. Therefore, an assembler is said to perform **isomorphic (one to one mapping) translation**.

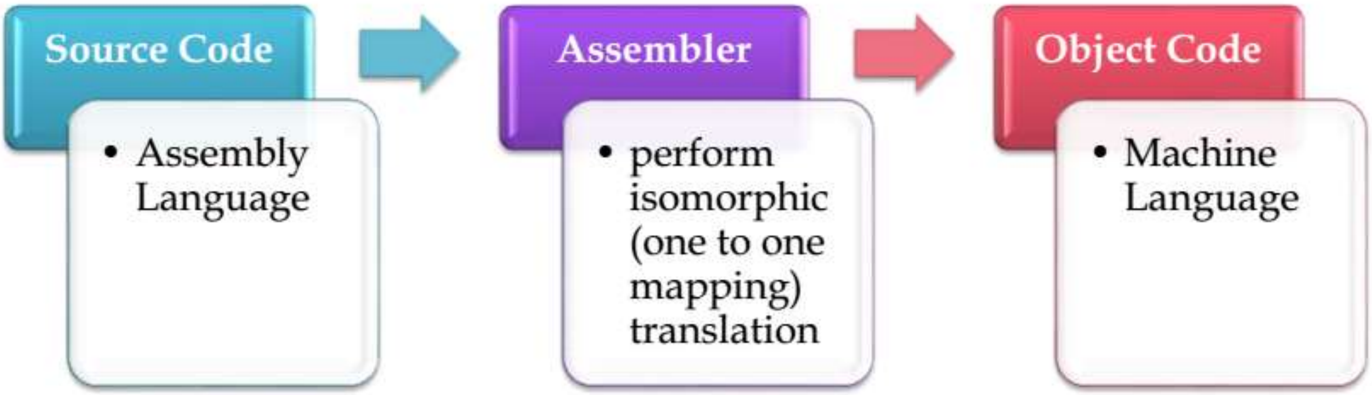


Fig. Assembler

Characteristic of A Good Programming Language

There are various factors, why the programmers prefer one language over the another. And some of very good characteristics of a good programming language are,

1) Clarity, Simplicity And Unity: A Programming language provides both a conceptual framework for Algorithm planning and means of expressing them. It should provide a clear, simple and unified set of concepts that can be used as primitives in developing algorithms.

It should have

•It has minimum number of different concepts
•with Rules for their combination being
•Simple and regular.

This attribute is called **conceptual integrity.**

2) Orthogonality: It is one of the most important feature of PL orthogonality is the property that means **" Changing A does not change B".**

If I take Real world example of an orthogonal system Would be a radio, where changing the station does not change the volume and vice versa.

When the features of a language are orthogonal, language is **easier to learn** and **programs are easier to write** because only few exceptions and special cases to be remembered.

3) Support for Abstraction:- There is always found that a substantial gap remaining between the abstract data structure and operations that characterize the solution to a problem and their particular data structure and operations built into a language.

4) Programming Environment: An appropriate programming environment adds an extra utility and make language to be implemented easily like

The availability of- Reliable- Efficient - Well documentation

Speeding up creation and testing by-special Editors- testing packages

Facility- Maintaining and Modifying- Multi Version of program software product.

Facility
• Maintaining
&
• Modifying
Multiple Version of program | Software P.

5) Ease of program verification: -

Reusability:

The reusability of program written in a language is always a central concern. A program is checked by various testing technique like

Formal verification method Desk checking Input output test checking.

We verify the program by many more techniques. A language that makes program verification difficult maybe far more troublesome to use. **Simplicity of semantic and syntactic structure is a primary aspect that tends to simplify program verification.**

6) portability of programs: Programming language should be portable means it should be easy to transfer a program from which they are developed to the other computer.

A program whose definition is independent of features of a Particular machine forms can only support **Portability. Example: Ada, FORTRAN, C, c++, Java.**

Practical Component: Drawing Flowcharts using Lucid Chart& Writing pseudo code(in latex optional) for the following problems

(i) Greatest of three numbers

(ii) Sum of N numbers

(iii) Computation of nCr

Computation of nCr

- Also called **Combinations**
- The combination is a method of selecting several items or symbols out of a larger group or a data set, where an order does not matter.
- It's usually represented by nCr
- Used to find different number of combinations C(n,r) of n distinct objects taken r at a time
- The Formula to find nCr, is:

$$C(n,r) = \frac{n!}{r!\,(n-r)!}$$

A factorial is the result of multiplying a given number of consecutive integers from 1 to the given number. It is written with the exclamation sign: n! and it is defined as

0! = 1

1! = 1

2! = 2 x 1 = 2

3! = 3 x 2 x 1 = 6

4! = 4 x 3 x 2 x 1 = 24

5! = 5 x 4 x 3 x 2 x 1 = 120 and so on.!

Assignment questions:

1. write the total configuration of your PC
2. Compare and contrast all generations of computers.
3. Assignment on Number system and conversions.
4. Assignment on algorithm, flowchart and pseudo code.

TWO MARKS

1. Define Computer.
2. What are the components of a computer?
3. What are the applications of a computer?
4. What are the characteristics of a computer?
5. Why computer is known as data processing system?
6. What is Data and Information?
7. Compare Computer with calculator
8. Differentiate analog and digital computers?
9. List down the hardware terms in all 5 generations of computers?
10. What are the languages used in computer generations?
11. Define high level language?
12. Define machine level language?
13. What is GUI?
14. What are the Classifications of computers?
15. List down the basic 4 units of computers?
16. What is the need of memory?
17. Difference between volatile and non volatile memory?
18. What are the primary memory available and its types?
19. Write short note on cache memory?
20. Define secondary memory with example?

21. Difference between RAM and ROM?
22. What are the 4 operations done by CPU?
23. What is the use of ALU and CU?
24. Write about functionalities of Control unit in CPU
25. What is meant by register and write different types register?
26. Why we need number system?
27. Write about input units and its types?
28. Write about output units and its types?
29. Expand the following: SSI, MSI, LSI, VLSI
30. Give the expansion for following: UNIVAC, ENIAC, EDVAC
31. Expand COBOL, BASIC, FORTRON and IBM
32. Define Clients and Servers?
33. Define Software and hardware?
34. What are the tasks performed by a processor unit in a computer?
35. What is an IC? How does it help in reducing the size of computers?
36. What are the positional Number systems?
37. Define Flowchart?
38. List down the flowchart symbols?
39. What are the rules available for drawing a flowchart?
40. What are the advantages and disadvantages of flowchart?
41. Differentiate algorithm, flowchart and pseudo code?
42. Mention the characteristics of algorithm?
43. Write an algorithm for calculating simple interest?
44. Write a Pseudo code for to check even or odd number?
45. Draw a flow chart for biggest of two numbers?

46. Find the decimal equivalent of hexadecimal number 4D.6F

47. Convert the number binary number 1101100111 to octal.

48. Write the binary and octal equivalent of hexadecimal number 7BD?

49. Convert hexadecimal number into binary equivalent of EBC.

50. Define Object Oriented Programming.

51. Define Logical Programming.

52. Define Procedural Programming.

53. Write about Low Level Language.

54. Write about High Level Language.

55. Write about Assembly Language.

8 MARKS:

1. Explain the characteristics of computer?
2. Draw a flowchart to find whether a given number is Armstrong or noT.
3. Draw a flowchart to find the sum of first 100 natural numbers.
4. Write an algorithm, flowchart and pseudo code for to check given number is palindrome or not
5. Write an algorithm, flowchart and pseudo code to find factorial of a number.
6. Explain the various types of computer memory
7. Explain about Flow Programming
8. Write briefly on Control Structures.
9. Explain about Object Oriented Programming
10. Write in brief on Generation of Programming Languages.

16 MARKS

1. Explain the Generations of computers in detail?
2. Explain in detail about evolution of computer?

3. Explain the classifications of computers?
4. Explain about Organization of Computers with neat block diagram?
5. Write an algorithm, flowchart and pseudo code to solve the quadratic Equation.
6. Write an algorithm, flowchart and pseudo code to find factorial of a number.
7. Explain the need for an algorithm and highlight its advantages. Write an algorithm to find the greatest among three numbers?
8. Convert the following:
    a) Convert (6245.14)8 to its decimal equivalent.
    b) Convert (111001.101)2 to its decimal equivalent.
    c) Convert the following numbers into their binary equivalent.
        - $(59.6825)_{10}$
        - $(EBC)_{16}$
        - $(654)_{8}$
9. Explain in detail on Programming Paradigms.
10. With Examples write in detail on Programming Control Structures.
11. With Examples write characteristics of good programming languages.