MICROPROCESSOR BASICS AND MEMORY CONCEPTS

MICROPROCESSOR:-

It is a multipurpose, programmable logic device (an Integrated Circuit) that accepts binary data as input and processes data according to those binary instructions reads from a storage device called memory and provides results as output. The basic characteristics of microprocessor are :-

(i) **Instruction Set**:-

The possible operations a microprocessor can execute, is determined through its instruction set. For example, SIMP(Single Instruction and Multiple Processor) computer understands 10 instructions and any program written for it uses those 10 instructions in various ways.

(ii) **Word Size**:-

The numbers of bits processed in single instruction depend upon the word size of a microprocessor. The word size determines the amount of RAM memory that can be accessed. 32-bit processor means it has a word size of 32 bits.

(iii) **Clock Speed**:-

It is the speed at which a microprocessor executes instructions. It is also called **clock rate**. The faster the clock, the more instructions the microprocessor can execute per second. Clock speed is stated as MHz(Mega Hertz) or GHz(Giga Hertz).

BUSES:-

It is a subsystem that concepts various computer components with CPU and transfers data/instructions between them. Figure below shown the connection:-

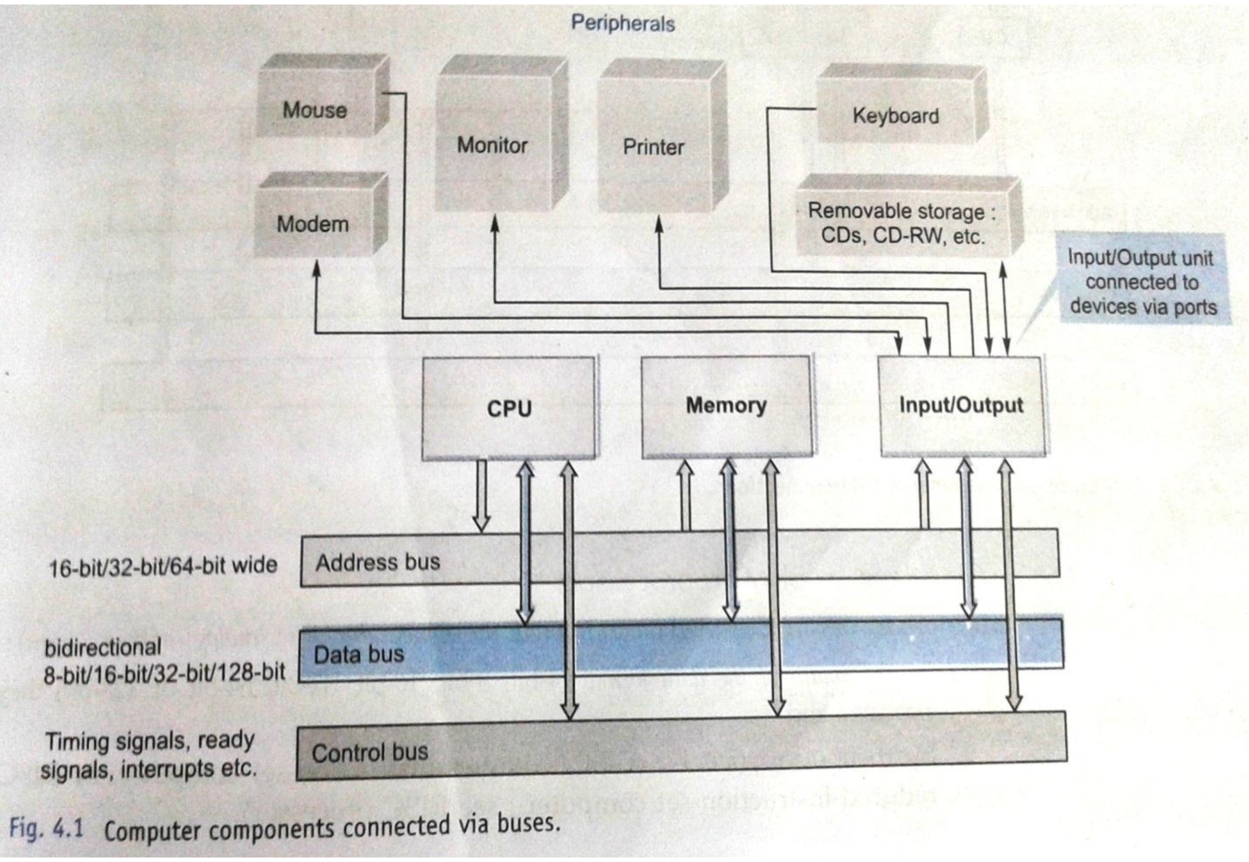

Peripherals
Mouse
Monitor
Printer
Keyboard
Modem
Removable storage : CDs, CD-RW, etc.
Input/Output unit connected to devices via ports
CPU
Memory
Input/Output
16-bit/32-bit/64-bit wide
Address bus
bidirectional 8-bit/16-bit/32-bit/128-bit
Data bus
Timing signals, ready signals, interrupts etc.
Control bus

Fig. 4.1 Computer components connected via buses.

There are following three types of busses that perform these tasks:-

(i) **Address bus** is unidirectional bus which is responsible for carrying address of a memory location or an input/output port from CPU.

(ii) **Data Bus** is bidirectional bus which carries the actual data to and from memory location, to output ports and from input ports depending on architecture i. e. 16 bits or 32 bits or 64 bits. These are two types:-

(a) **Internal Data Bus** that operates only within the internal circuitry of the CPU, communicating among the registers and internal caches of memory that are part of the CPU chip's design.

(b) **External Data Bus** that connects a computer to Main Memory and peripheral devices via ports.

(iii) **Control Bus** is a bidirectional bus that carries control signals such as clock signals; interrupt signals (ready, halt etc.).

Memory Bottleneck:-

In the above figure, it is clear how CPU is connected to memory and Input-Output unit via buses. Whenever CPU needs to work on data, it gives the address of corresponding memory location in RAM or address of input port on address bus. The data bus then fetches the data from specified location in RAM or from input port to CPU. This process is slow and it makes possible that CPU could have wait for data. This problem is called **Memory Bottleneck.**

The **simple solution** of this problem include:-

(i) **Registers** are attached within CPU itself to hold data currently in use.

(ii) Add a faster memory called **cache memory** that holds frequently used data. These new memory are connected via internal data buses.

With solution of memory bottleneck problem, the new microprocessor component and their connection look like as shown below:-

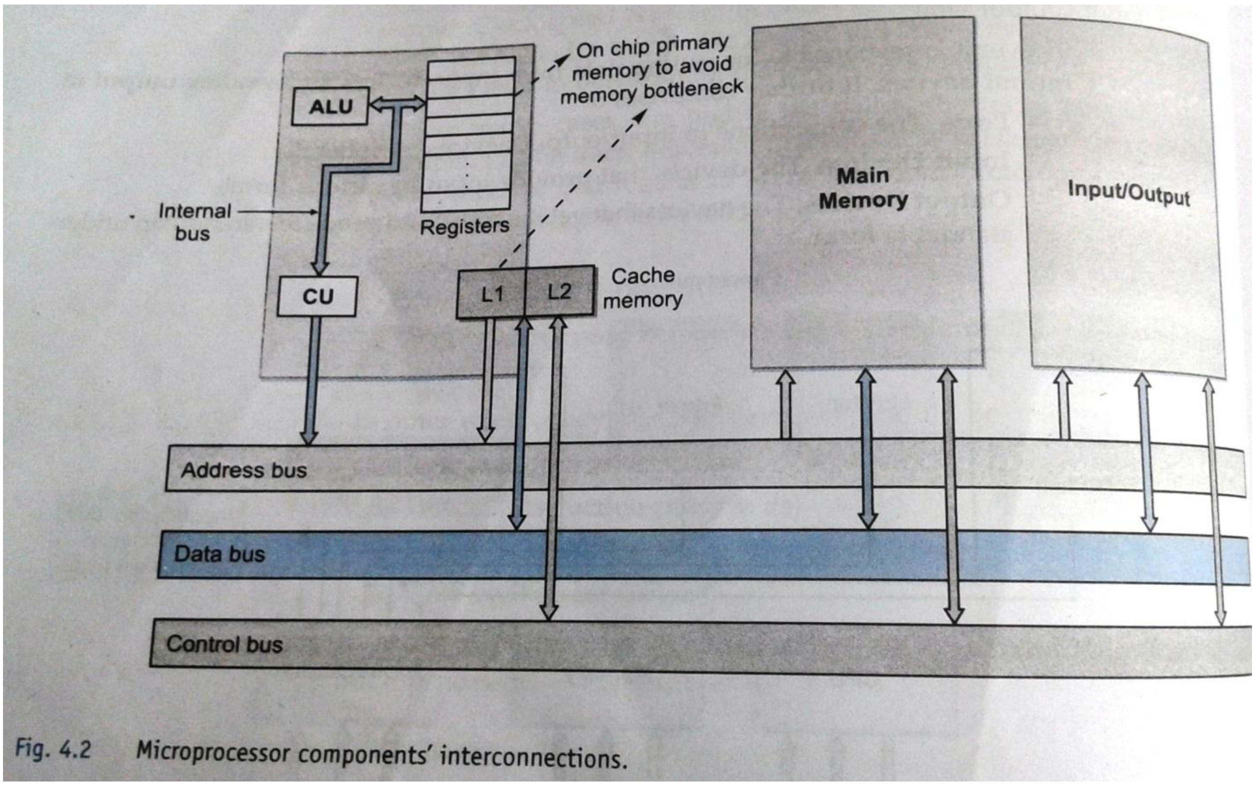


Fig. 4.2 Microprocessor components' interconnections.

Classification of Microprocessors:-

Microprocessors are classified :-

(i) **By the width of the data format they process:-**

This type of microprocessors depends upon the size of the internal registers and the word-size of the data-width they can process in one go. For example, 4-bit, 8-bit, 16-bit, 32-bit, 64-bit or 128-bit.

(ii) **By their instruction set:-**

(a) RISC (Reduced Instruction Set Computer) have the following characteristics:-

- Simple primitive instructions and addressing modes
- Instructions execute in one clock cycle
- Uniformed length instructions and fixed instruction format
- Hardwired control
- Instructions interface with memory via fixed mechanisms
- Complexity pushed to the compiler
- Supports fewer instructions
- For example, Pentium1, AMD K6, Intel P6 etc.

(b) CISC (Complex Instruction Set Computer) have the following characteristics:-

- Supports hundreds of instructions
- Richer instruction set, some simple, some very complex
- Instructions generally take more than 1 clock to execute
- Instructions of variable sizes
- For example, Intel 386, Intel 486, Pentium III, Pentium Pro etc.

(c) EPIC (Explicitly Parallel Instruction Computing) have the following characteristics:-

- It offers best of both above computers.
- It provides parallelism
- It does not use fixed width instructions rather as many as parallel as possible
- Programs must be written using sequential semantics with explicitly laid out parallelism
- Compiler should play the key role in designing the plan of execution and the architecture should provide the requisite support for it to do so successfully.

Memory Devices:-

Memory is an essential component of the computer system. A high speed memory is organized into words of fixed lengths. A memory is divided into N words where each word is assigned an address in the memory. Each word consists of same number of bits is called **word length**. A memory with 4096 locations with each location having 16 bits is called a 4096 word 16-bit memory or 4 K 16-bit memory (1K = 1024). Memory is divided into following categories:-

(i) Main Memory or Primary Memory:-

Programs and Data need to be in main memory in order to be executed or referenced. The main Memory can be divided into two parts:-

(a) RAM (Random Access Memory):-

It is a temporary memory or volatile memory whose data would be erased when the system has been shut down or when the power goes off.

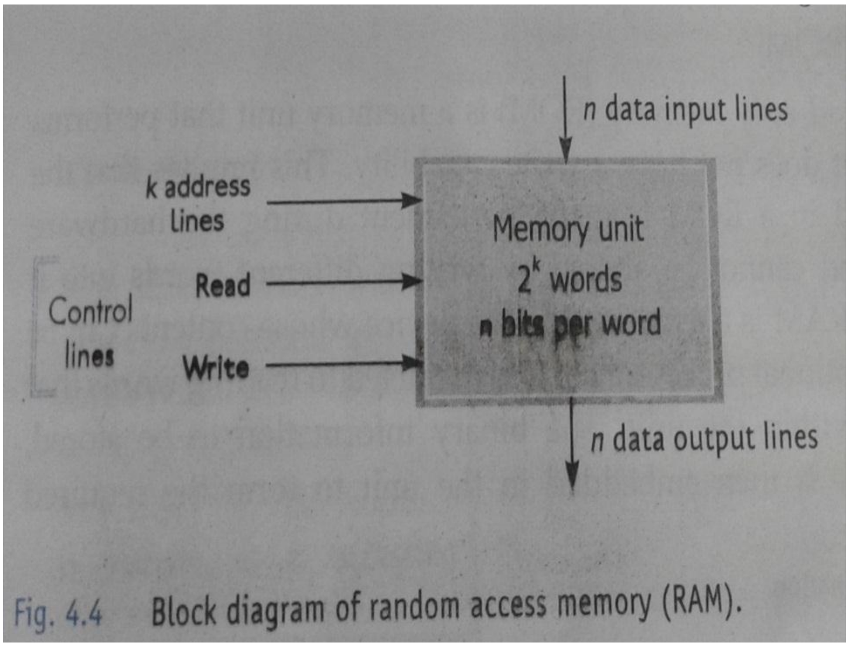


Fig. 4.4 Block diagram of random access memory (RAM).

The k address lines provide a binary number of k bits that specify the address of a particular word chosen among the 2^k available inside the memory. The RAM can be of two basic types:

- **Dynamic RAM(DRAM)**:-

  It consists of a transistor and capacitor that's capable of storing an electric charge.

  The amount of time taken to produce data required from memory, from the start of access until the availability of data is called **memory access time**.

  A DRAM cell consists of only one transistor and capacitor per bit, it allows a DRAM chip to pack a large number of cells within the chip compared to SDRAM.

  The problem of a capacitor is that it starts loosing the charge over a period of time, and can retain data for barely a thousandth of a second. Therefore, the memory controller needs to refresh the memory contents as many as thousand times a second, which is called **memory refreshing.**

- **Static RAM**:-

  It consists of internal flip-flops that store the binary information. It take up more space for a given storage capacity than do dynamic RAMs. Static RAMs are used in specialized applications while dynamic RAMs are used in the primary storage sections of most computers. The static RAM is easier to use and has shorter read and write cycles compared to dynamic RAM.

(b) ROM (Read Only Memory):-

It is the memory that performs the read only operation. It does not have write capability. There are various types of ROM given below:-

- **Programmable Read Only Memory (PROM)** is a ROM that can be programmed to record information using a facility known as PROM-programmer that information cannot be changed.
- **Erasable Programmable Read Only Memory (EPROM)** is another type of ROM that can be erased using **ultraviolet light (UPROM)** & **Electrically Alterable (EAPROM)** and the chip can be reprogrammed to record different information using a special PROM-program facility.
- **Electrically Erasable PROM (EEPROM)** is ROM that can be programmed and erased by electrical signals. Information loaded in this memory can be retained for many years without power supplied, when power is returned, the EEPROM memory can be used to replace the lost contents of the RAM memory and the microcomputer can continue working just as if nothing had happened.

(ii) Cache Memory:-

The Cache Memory is high speed memory available inside CPU in order to speed up access to data and instructions stored in RAM memory. 95% of the time the processor is working; it is accessing information from cache. When data is found in the cache, it is called a **cache hit**, and the effectiveness of a cache is judged by its hit rate. There are three types of cache memories:

- **Level 1 (L1) cache** is built inside the CPU package
- **Level 2 (L2) cache** is external to the CPU and reside on the motherboard.
- **Level 3 (L3) cache** is external to the CPU and reside on the motherboard.

Thus, when CPU needs some information, its order of memory checks or access as shown below:

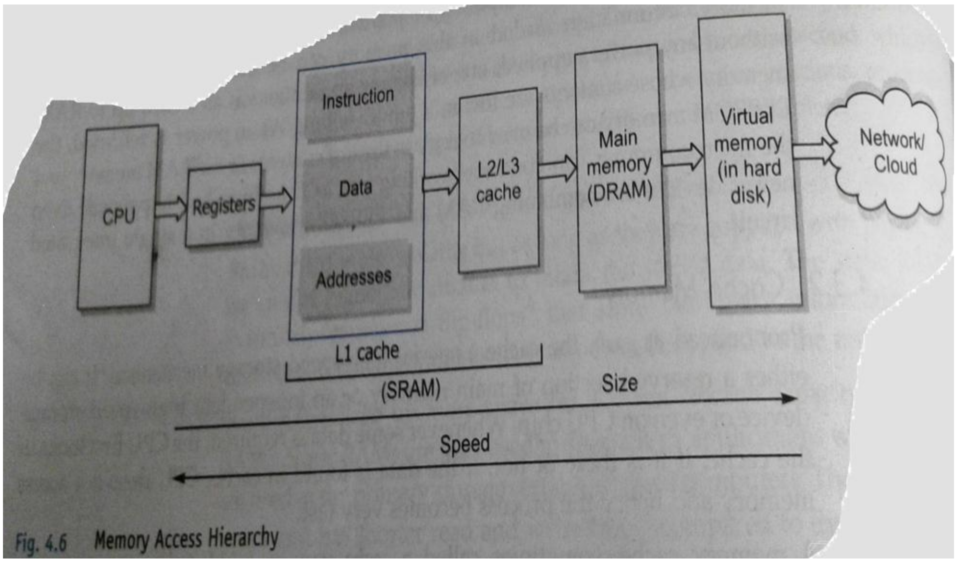


Fig. 4.6 Memory Access Hierarchy

(iii) Secondary Memory:-

These are also called Secondary Memory Devices or Storage Devices which are used to store large amount of data permanently. Some most common storage devices are:-

- **Floppy Disks or Diskette**

- **Hard Disks**:- These are the memories store information on one or more circular platters or disks which are continually spinning. These rotating disks are coated with a magnetic material and stacked with space between them. Information is recorded on the surface of rotating disks by magnetic heads as tiny magnetic spots. These heads are mounted on access arms. Information is recorded in bands. Each band of information on a given disks is called a **track** i.e. the concentric circles on the magnetized surface of the magnetic disks **are known as tracks**. The tracks are commonly divided into invisible pie-shaped sections called **sectors**.
- **Compact Disks (CDs)**:- these are optical media storage capacity of upto 700 MB. There are three main types:-
  - **(a) CD-ROM (CD-Read Only Memory)** used only to store information and cannot be used to store data. Manufacturers use CDROMs to record information for the CD distribution e.g. encyclopedias, software, games, e-books etc.
  - **(b) CD-R (CD-Recordable)** can be written data only once and disk can't be erased.
  - **(c) CD-RW (CD-Rewritable)** is an erasable disk that you can write on multiple times.
- **Digital Video Disks (DVDs)** is an optical storage device that store information about 15 times as much information and transfer it to the computer about 20 times as fast as a CD-ROM. It is also called **Super Density disk (SD)**. It capable of storing 4.7 GB to 17 GB. There are three main types:-
  - **(a) DVD-ROM (DVD-Read Only Memory)**
  - **(b) DVD-R (DVD -Recordable)**
  - **(c) DVD-RW (DVD-Rewritable)**
- **Pen/Thumb Drives – Flash Memories**
- **Blu-ray Disk (BD)** is an optical disc storage media format. Its main uses are high-definition video and data storage. It capable to hold 25 GB/50 GB almost six times the capacity of a dual layer DVD.

PORTS:-

(i) **Serial Ports:-** these ports transfers data serially a bit at a time. As a result, the serial port needs only wire to transmit 8 bits. Serial ports are often known as communication (COM) ports or RS232C ports.

(ii) **Parallel Ports:-** these ports can send or receive a byte (8-bit) at a time. It comes in the form of 25-pin female connector.

(iii) **USB (Universal Serial Bus) Ports**:- To spare the user botheration of 8-pin, 25-pin, male, female connectors, the USB has been designed.

(iv) **AGP Ports:-** it is used to connect to graphic card that provides high-speed video performance typically required in games and other multimedia applications.

(v) **InfraRed Port (IR Port):-** It is a port, which sends and receives infrared signals from other devices. In this special form of radio transmission, a focused ray of light in the infrared frequency spectrum, measured in terahertz, or trillions of hertz (cycles per second). It is playing important role in wireless data communication.

(vi) **Bluetooth** is a telecommunication industry specification that describes how mobile phones, computers and personal digital assistants (PDAs) can be easily interconnected using a short-range wireless connection. Connections can be point-to-point or multipoint. The maximum range is 10 meters.

(vii) **PS-2 Port** stand for Personal System/2. It is an electronic receptacle or plug found on computers. It accepts a PS/2 cable (Female) with a mini-DIN connector (Male) and is most often used to plug in a keyboard or mouse. DIN stands for standards setting organization for Germany.

(viii) **Firewire** is Apple Computer's version of a standard, IEEE 1394, High Performance Serial Bus, for connecting devices to your personal computer. It provides a single plug-and-socket connection on which upto 63 devices can be attached with data transfer speeds upto 800 Mbps(Megabits per second).

General OOP Concepts

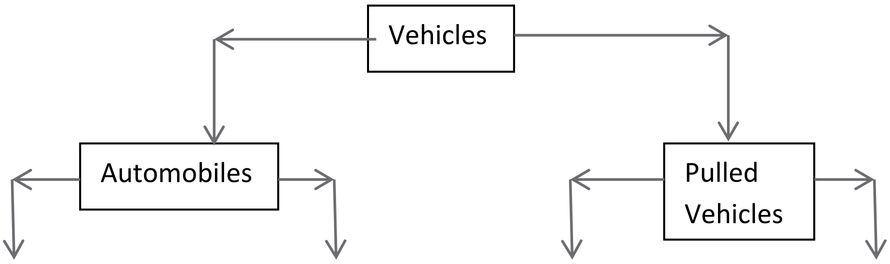
Paradigm

It means organizing principle of a program. It is an approach to programming. It means a way of thinking or doing things. The different programming paradigms are:-

1. **Procedural Programming** is emphasis on doing things. It is very time-consuming and frustrating for large-sized programs. It does not model real world very well.
2. **Modular Programming** is an extension of procedural programming. It focuses on the procedures or the working action. In this programing, a big program is divided into smaller parts known as modules. A **Module** is an identifiable piece of code within a program, with a set of inputs and outputs. It has a sole purpose and processes its inputs in a way dictated by its goal and provides some output. These modules work independently of all others.
3. **Object Oriented Programming** is a method of implementation in which programs are organized as cooperative collections of objects, each of which represents an instance of some class and whose classes are all members of a hierarchy of classes united via inheritance relationship where:
    - **Object** is an identifiable entity with some **characteristics** and **behavior**. For example, orange is an **object**. Its **characteristics** are it is spherical shaped, its color is orange etc. Its **behavior** is it is juicy and it tastes sweet-sour.
    - **Class** is group of objects that share common properties and relationships. We can say that the wrapping up of data and its associated operations/functions that operate on the data into a single unit, this unit is called class. For example, car is a class and optra, nano, wagnor etc. are objects.

Basic Concepts of OOP

1. **Data Abstraction** refers to the act of representing essential features without including the background details or explanations. For example, to drive a car you have to know about internal details of car such as gear handling, braking, accelerator, brakes etc. but while driving these internal details are hidden from you.
2. **Data Encapsulation** refers that the wrapping up of data and operations/functions that operate on the data into a single unit, this unit is called class and the process to build a class is known as encapsulation. It is a way to implement data abstraction. It hides the details of the implementation of an object. For example, car.
3. **Modularity** is the property of a system that has been decomposed into a set of cohesive and loosely coupled modules. The justification for portioning a program is that it reduces its complexity to some degree and creates a number of well-defined, documented boundaries within the program. For example, music system comprises of speakers, cassette-player, record-player etc.
4. **Inheritance** is the capability of one class of things to inherit capabilities or properties from another class. For example, figure as shown below:-



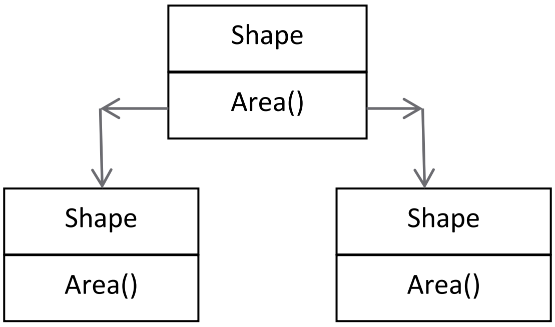
Car
Bus
Cart
Rickshaw

Where vehicles inherited by automobiles and pulled vehicles and so on. Vehicles is the base class or super class and Automobiles & Pulled Vehicles are the derived classes or sub classes for Vehicles base class.

Reasons for the introduction of inheritance are:-

- It has the capability to express the inheritance relationship which makes it ensure the closeness with the real-world models.
- Idea of reusability
- Transitive nature of inheritance means if a class **A** inherits properties of another class **B**, then all subclasses of **A** will automatically inherit the properties of **B**. this property is called transitive nature of inheritance.

5. **Polymorphism** is the ability for a message or data to be processed in more than one form. Languages that support classes but not inheritance and polymorphism are called object-based languages. For example, as shown below:



Advantages and Disadvantages of OOPs:-

Advantages:

- Re-use of Code
- Ease of comprehension
- Ease of fabrication and maintenance
- Easy redesign and extension

Disadvantages:

- Classes tend be overly generalized
- Design is tricky
- Need proper planning and proper design for OOP programming
- Need proper skills such as design, programming, thinking in terms of objects etc.

GETTING STARTED WITH C++

C++ Character Set:-

Character set is a set of valid characters that a language can recognize. A character represents any letter, digit, or any other sign. The C++ has the following character set:-

• Letters	A to Z, a to z
• Digits	0 to 9
• Special Symbols	
• Other characters	

Tokens (Lexical Units):-

A program is divided into small individual unit such as individual words, punctuation marks etc. are called tokens or lexical units or lexical elements. C++ has the following tokens:-

1. **Keywords:-**

   These are the words having special meaning reserved by programming language for e.g. – int, void, main, include, const, cin, cout etc.

2. **Identifiers:-**

   These are the basic building blocks of a program and are used as the general terminology for the names given to different part of program such as variables, objects, classes, function, arrays etc.

   To form an identifier, C++ follow the basic rules:-

   (i) It is a sequence of letters and digits.
   (ii) The 1st character must be a letter.
   (iii) The underscore (_) counts as a letter.
   (iv) Upper and lower cased letters are different, that means C++ is a case sensitive language.
   (v) Keywords cannot be identifier.
   (vi) For example:-

Valid identifies:-	Myfile	_abhist	ak234mK
Invalid identifiers:-	2foRti	break	My.file

3. **Literals:-**

   Literals often referred to as constants are data items that never change their value during a program run. Literals can be several kinds:-

   (i) Boolean
   (ii) Integer
   (iii) Character
   (iv) Floating
   (v) String

   For e.g.- const int a = 10;   const float c = 3.4;
   const char b = A;   const char e[ ] = “Mukesh”;

   - **Nongraphic characters or Escape Sequences:-**

     Character can be nongraphic or escape sequences that cannot be typed directly from keyboard e.g., \n, \a, \b, backspace, tabs, carriage return etc.

4. **Punctuators:-**
   These are special symbols which are used to separate the different parts of the program.
   For example:- { } ( ) [ ] # , ; : * etc.
5. **Operators:-**
   Operators are tokens that trigger some computation when applied to variables and other objects in an expression. The operators are of following types:-
   (i) Unary Operators:-
   Those operators that require one operator to operate upon. Some unary operators are &, *, +, -, ~, ++, --, !
   (ii) Binary Operators:-
   Those operators that require two operands to operate upon. Some binary operators are :-
   (a) Arithmetic operators:- Addition(+), Subtraction(-), Multiplication(*), Division(/), Remainder/Modulus(%)
   (b) Logical operators:- Logical AND(&&), Logical OR(||)
   (c) Assignment operators:- Assignment(=), Assign quotient(/=), Assign sum(+=), Assign product(*=), Assign remainder(%=), Assign difference(-=)
   (d) Relational operators:- Less than(<), Less than equal to(<=), Greater than(>), Greater than equal to(>=), Equal to(==), Not equal to(!=)
   (iii) Conditional Operator(?:)
   (iv) Scope Access/Resolution Operator(::)

Stream:-

It is simply a sequence of bytes. Input and Output operations are supported by the **istream** and **ostream classes**. The predefined stream objects for input and output are **cin** and **cout**. Declarations and functions of **cin** and **cout** are contained within **iostream.h**.

Comments in a C++ Program:-

Comments are pieces of code that the compiler discards or ignores or simply does not execute. The purpose of comments is only to allow the programmer to insert some notes or descriptions to enhance readability or understandability of the program. There are two ways to insert comments in C++ programs:-

1. **Single Line Comments with //.**
2. **Multiline or block comments with /*………..*/**

Output Operator (<<)(“put to”):-

It is also called stream insertion operator is used to direct a value to standard output. For example,

```
cout<<”The sum of 2+5=”;
cout<<2+5;
cout<<”\n”;
```

Input Operator (>>)(“get from”):-

It is also called stream extraction operator is used to direct a value to standard output. For example,

```
cin>>value1;
cin>>manjit;
```

where value1 and manjit are identifiers or variables.

Cascading of I/O Operators:-

The multiple use of input or output operators (>> or <<) in one statement (of cin or cout) is called cascading of I/O operators. For example,

```
cout<<"The sum of 2+5="<<2+5<<"\n";
cin>>value1>>manjit;
```

Role of Compiler:-

Compiler converts the High level language program into object code (machine language program) in one go. It analyse the program code for “correctness”. Compiler reports an error by flashing an error message. An error message contains a line number and a brief description of the error. After correcting the program, the program should be recompiled and retested. It translates the corrected program text into object or assembly instruction text understood by the computer. This process of translation is called code generation. After code generation the actual program execution takes place. Some common errors are:-

1. **Syntax Error:-**
   These errors occur when rules of a programming language are misused i.e. when a grammatical rule of C++ is violated. For example,
   ```
   cin<<value1;
   cout>>"The sum of 2+5=";
   ```
2. **Semantics Error:-**
   These errors occur when statements are not meaningful. For example,
   ```
   X * Y = Z;
   ```
3. **Type Errors:-**
   These errors occur when data/value of unexpected type is passed or input.
4. **Run-time Errors (Execution Errors):-**
   These errors occur during execution of program i. e. during run time generally when some resources is unavailable. For example, “File not found”, “Out of Memory”, Divided by zero” etc.
5. **Logical Errors:-**
   These errors occur when a program to produce incorrect or undesired output. For example,
   ```
   avg = a * b * c / 3;    //intention is to compute average but result is incorrect
   sum = a – b;            //intention is to compute addition but result is incorrect
   ```

Operators & Expressions in C++

Operators:-

The operations (specific tasks) are represented by operators and the objects of the operation(s) are referred to as Operands. There are number of operators used in C++:-

1. **Arithmetic Operators:-**

   These operators perform arithmetic operations:-

   (i) **Unary Operators**:- Operators that cat on one operand are referred to as Unary Operators. For example, if a = 5 then +a means 5 and –a means -5.

   (ii) **Binary Operators**:- Operators that act upon two operands are referred to as Binary Operators. For example, 4 + 5, a-7, a*5, a/b, a%b.

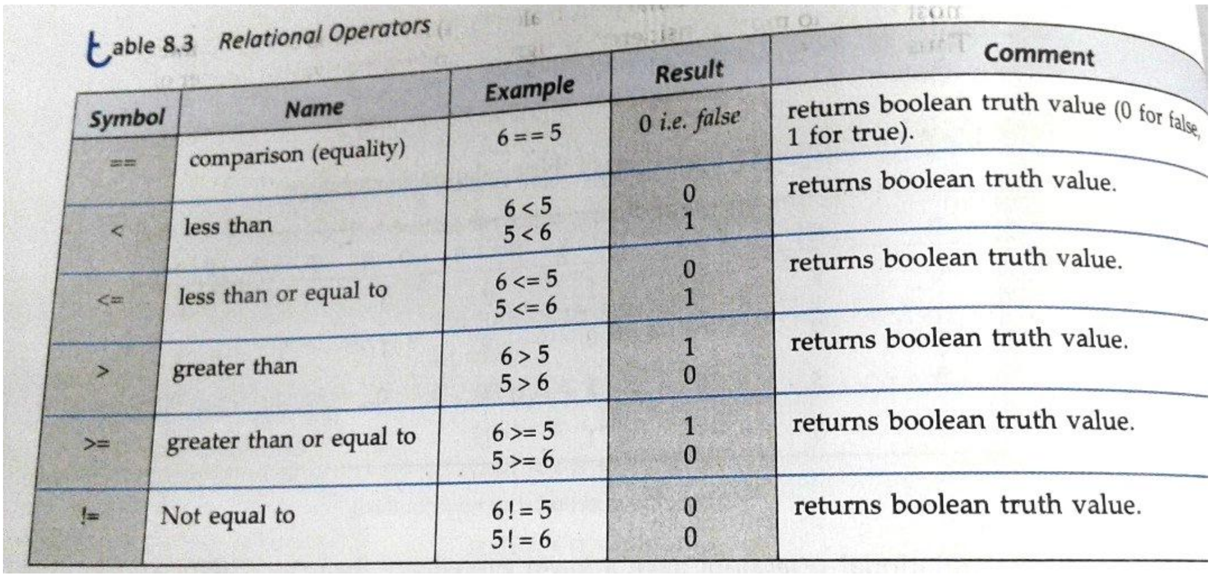
2. **Increment/Decrement:-**

   These are the increment/decrement operators , ++ and --. The operator ++ adds 1 to its operand and – subtracts 1. For example, a = a + 1, is the same as ++a; or a++; a = a – 1, is same as –a; or a--;

   The prefix increment of decrement operators follow **change then use** rule. The postfix increment of decrement operators follow **use then change** rule.

3. **Relational Operators:-**

   The relational operators determine the relation among different operands. **Any non-zero value is true in C++**. **Only zero(0) considers as false**.

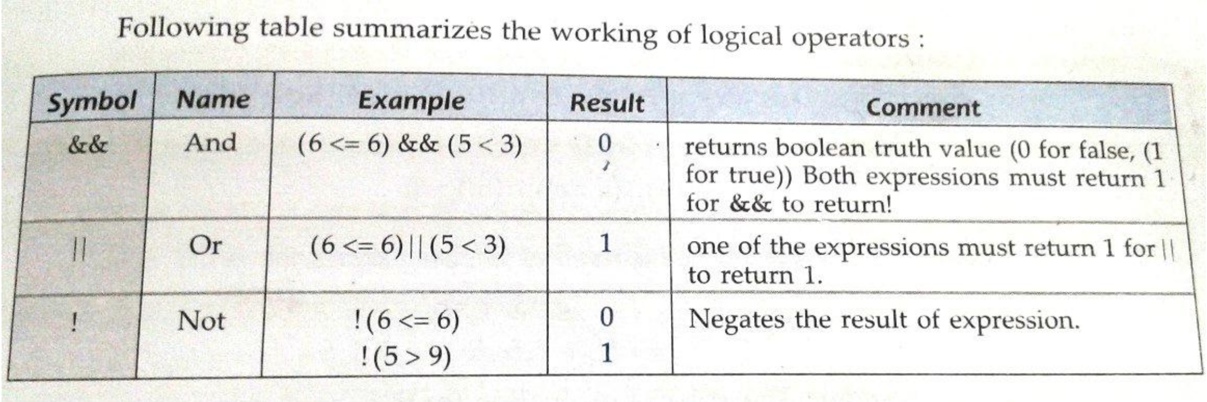


able 8.3 Relational Operators

Symbol	Name	Example	Result	Comment
==	comparison (equality)	6 = = 5	0 *i.e. false*	returns boolean truth value (0 for false, 1 for true).
<	less than	6 < 5 5 < 6	0 1	returns boolean truth value.
<=	less than or equal to	6 <= 5 5 <= 6	0 1	returns boolean truth value.
>	greater than	6 > 5 5 > 6	1 0	returns boolean truth value.
>=	greater than or equal to	6 >= 5 5 >= 6	1 0	returns boolean truth value.
!=	Not equal to	6 ! = 5 5 ! = 6	0 0	returns boolean truth value.

4. **Logical Operators:-**

   C++ provides three logical operators to combine existing expressions. These are ||(Logical OR), &&(Logical AND) and !(Logical NOT).



Following table summarizes the working of logical operators :

Symbol	Name	Example	Result	Comment
&&	And	(6 <= 6) && (5 < 3)	0	returns boolean truth value (0 for false, (1 for true)) Both expressions must return 1 for && to return!
||	Or	(6 <= 6) || (5 < 3)	1	one of the expressions must return 1 for || to return 1.
!	Not	! (6 <= 6) ! (5 > 9)	0 1	Negates the result of expression.

5. **Conditional Operator:-**

   C++ offers a conditional operator(?:) that stores a value depending upon a condition. This operator is ternary operator i.e., it requires three operands. The general form is:-

   Expression1? Expression2 : Expression3

   For example,

   Result = marks>=50? ”Pass” : “Fail”;

   Bonus = n + sales > 15000 ? 250 : 50;

6. **Sizeof operator:-**

   It is compile-time operator that returns the length (in bytes) of the variable or parenthesized type-specifier that it precedes. It can be used in two forms as:

   sizeof(type)   where type is a C++ data type

   OR

   sizeof(var)   where var is a declared variable

7. **Comma Operator:-**

   It is used to string together several expressions. The group of expressions separated by commas (,) is evaluated left-to-right in sequence and result of the right-most expression becomes the value of the total comma-separated expression. For example,

   B = (a = 3, a+1, a-5, a*4);   then B=12;

Precedence of Operators (Order of Precedence):-

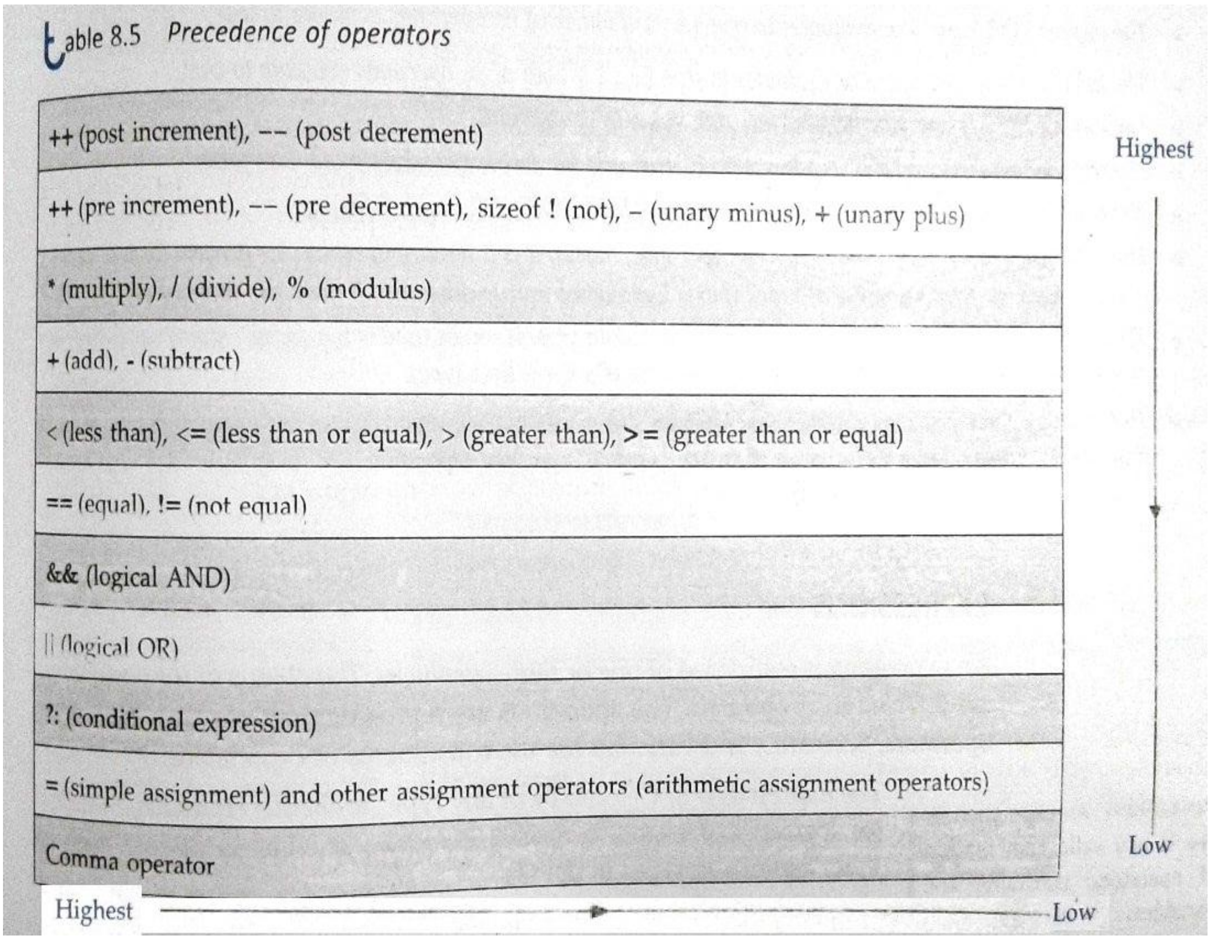


Table 8.5 *Precedence of operators*

++ (post increment), -- (post decrement)	Highest
++ (pre increment), -- (pre decrement), sizeof ! (not), – (unary minus), + (unary plus)	
* (multiply), / (divide), % (modulus)	
+ (add), - (subtract)	
< (less than), <= (less than or equal), > (greater than), >= (greater than or equal)	
== (equal), != (not equal)	
&& (logical AND)	
|| (logical OR)	
?: (conditional expression)	
= (simple assignment) and other assignment operators (arithmetic assignment operators)	
Comma operator	Low

Highest ——► Low

Type Conversion:-

The process of converting one predefined type into another. C++ facilitates the type conversion in two forms:

(i) **Implicit type conversion or type promotion**:-

It is a conversion performed by the compiler automatically without programmer's intervention. The C++ compiler converts all operands upto the type of largest operand, which is called **type promotion.**

(ii) **Explicit type conversion or type casting**:-

It is user-defined conversion that forces an expression to be of specific type conversion. It is also known as **type casting**. Type casting in C++ is done as:

(type) (expression)   where type is data type

For example,

(float)(x+y/2)

Type Compatibility:-

The value of the right side (expression side) of the assignment is converted to the type of the left side (target variable). For example,

Ch = x;   (where Ch is char and x is int)

C++ Shorthands:-

C++ offers special shorthands that simplify the coding such as:-

A = A + 10;	CAN BE WRITTEN AS	A + = 10;
A = A + B;	CAN BE WRITTEN AS	A + = B;
A = A - B;	CAN BE WRITTEN AS	A - = B;
A = A * B;	CAN BE WRITTEN AS	A * = B;
A = A / B;	CAN BE WRITTEN AS	A / = B;
A = A % B;	CAN BE WRITTEN AS	A % = B;