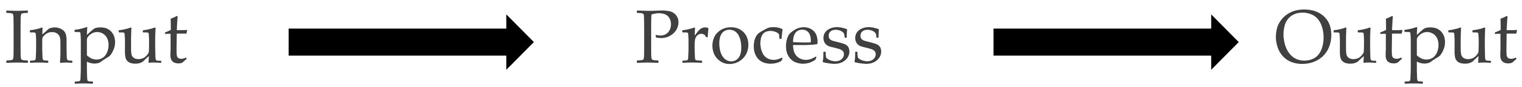
Computer Fundamentals

BY JANHAVI DEO

What is a Computer ?

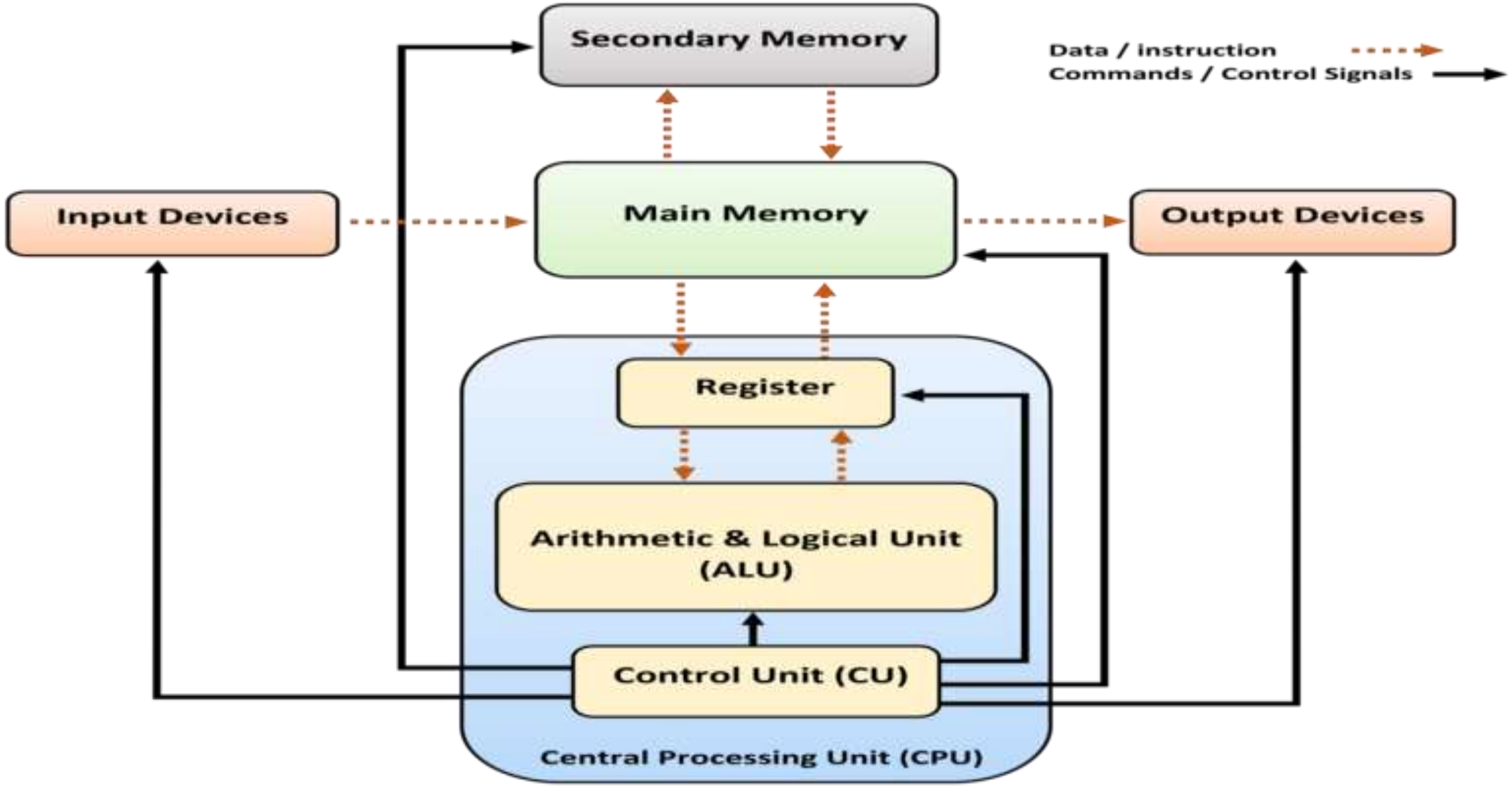
What is a Computer ?

- A computer is an electronic device (machine) that takes data and instructions and performs computations on data based on those instructions.
- Here computations means some kind of process.



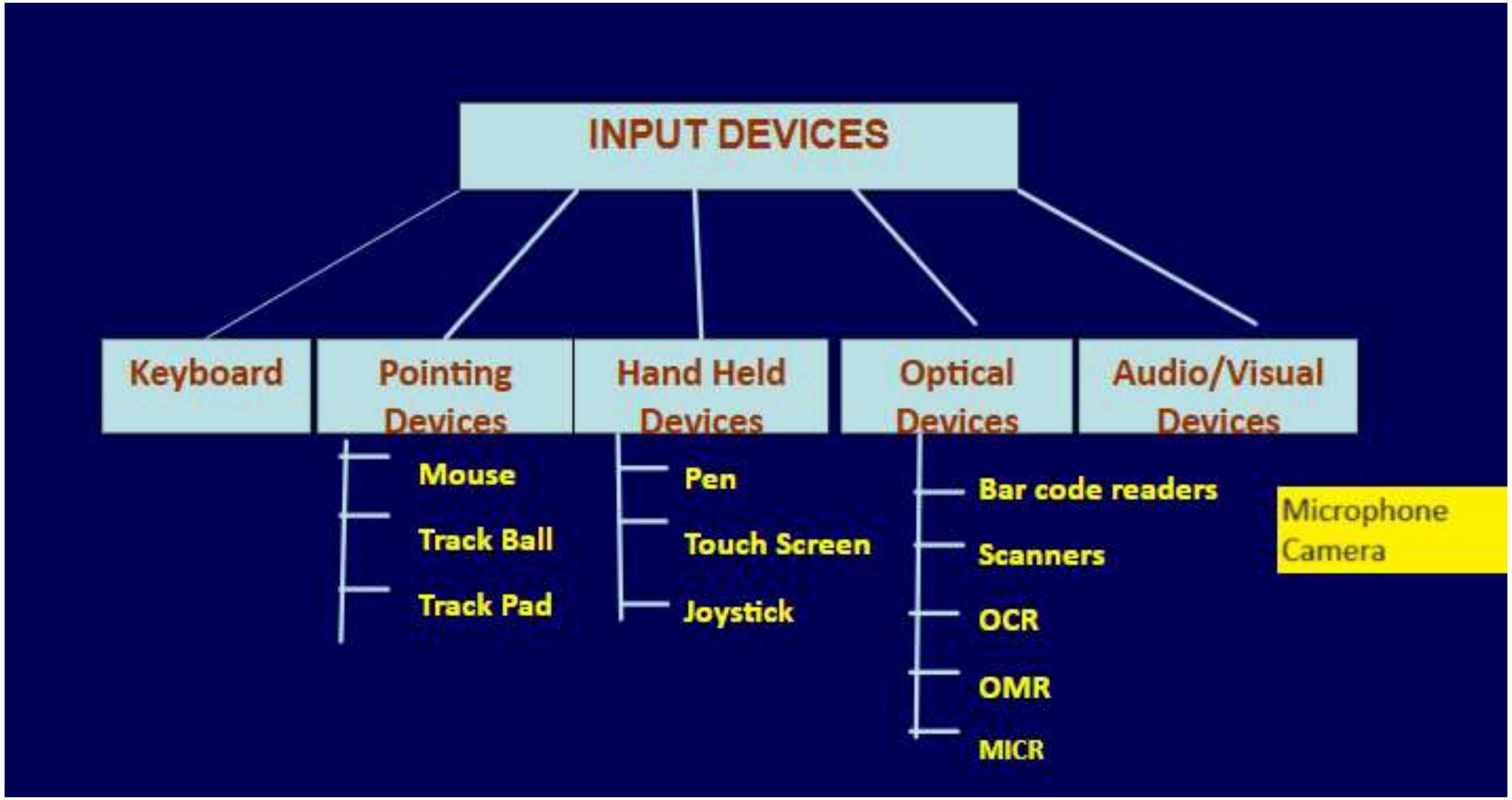
Input → Process → Output

Computer Block Diagram



Input Devices

Input Devices



Output Devices

- Monitor
- Printer
- Plotter
- Headphone
- Projector
- Speaker
- Touch screen
- LCD monitors

Computer generations

The evolution of technology to distinguish the computers in terms of changing hardware and software.

The 5 types of generation of computer are :

1st generation: (1940-1956) Vacuum Tubes

The first generation computers used very large number of vacuum tubes for circuitry and magnetic drums for memory.

UNIVAC and ENIAC computers

Second Generation (1956-1963) Transistors:

- The second generation computers were manufactured using transistors.
- Assembly languages introduced, which allowed programmers to specify instructions in words closer to English.
- High-level programming languages like COBOL, FORTRAN, ALGOL and SNOBOL were also being developed.

Third Generation (1964-1971) Integrated Circuits

- The development of the integrated circuit was the major turning point of the third generation of computers.
- High level programming languages like COBOL and FORTRAN were standardized.
- Some more high level programming languages like PASCAL and BASIC were introduced at this time.
- Third generation computers were the first to implement time sharing operating systems. Input could now be provided using keyboards and mouse.

Fourth Generation (1971-1989) Microprocessors

- The microprocessor started the fourth generation of computers with thousands of integrated circuits built onto a single silicon chip.
- Semi-conductor memories were used which were very fast, even the hard disks became cheaper, smaller in size and larger in capacity.
- For input, floppy disks (in addition to magnetic tapes) were used.
- During this period many new operating systems were developed like MS-DOS MS-Windows UNIX and Apple's proprietary operating system.
- Development of GUIs, the mouse and handheld devices.
- In this period, several word processing packages, spreadsheet packages and graphics packages were introduced.

Fifth Generation (Present and Beyond) Artificial Intelligence

- The fifth generation computers are completely based on a new concept of artificial intelligence.

- Although such computers are still in development, there are certain applications like voice recognition which is widely being used today.

- In the fifth generation of computers the aim is to develop devices that respond to natural language input and are capable of learning and self-organization.

- The two most common languages are LISP and Prolog.

Computer Generations and Languages

I generation : Machine level language. It consists of 0s and 1s. No translator used to translate the program. The instructions were entered through front panel switches of the computer system.

II generation : Assembly language and it belongs to machine level language. The language is specific to a particular processor family and environment. This is used for developing parts of kernel and device drivers.

Computer Generations and Languages

III generation : is a high-level computer programming language that tends to be more machine-independent and programmer-friendly than the machine code of the first-generation and assembly languages of the second-generation.

Fortran, ALGOL, and COBOL are examples of early 3GLs.

Most popular general-purpose languages today, such as C, C++, C#, Java, BASIC and Pascal, are also third-generation languages. Most 3GLs support structured programming. Many support object-oriented programming also.

Computer Generations and Languages

IV generation : Identified as "**non-procedural**" or "**program-generating**" languages. 4GL are more focused on databases, reports, and websites development.

Unix Shell, SQL, PL/SQL, Oracle Reports, R

Computer Generations and Languages

V generation : Fifth-generation languages are designed to make the computer solve a given problem without the programmer.

Google : List of programming languages for artificial intelligence

check : List of programming languages for artificial intelligence - Wikipedia

Binary Number System

Computer is an electronic device which uses electronic circuits

Each circuit can hold two different values to represent the two voltage levels (logical 0 for 0V and logical 1 for +5V)

The two values 0 and 1, corresponds to the two digits used by the binary number system.

Some important terminologies

Bit: Bit is a short form of binary digit. It is the smallest possible unit of data. In computerized data a bit can either be 0 or 1.

Nibble: Nibble is a group of four binary digits.

Byte: Byte is a group of eight bits.

Word: A group of two bytes is called a word. computers today have redefined word as a group of 4 bytes (32 bits) or 8 bytes (64 bits)

Some points to discuss

Convert a Decimal number into a binary number

Convert a Binary number into a decimal number

Octal number system and conversion

Hexadecimal number system and conversion

Encoding schemes

BINARY CODED DECIMAL (BCD) CODE

BCD is a technique of encoding decimal numbers in which, each digit is represented by its own binary sequence.

American Standard Code for Information Interchange (ASCII) Codes

It is a 7-bit character code used by personal and workstation computers

ASCII is the most common format for text files in computers and on the Internet. It can define 128 different characters because it is a 7 bit code.

Extended Binary Coded Decimal Interchange Code (EBCDIC)

An 8-bit character encoding technique used on IBM mainframes. EBCDIC supports a wider range of control characters than ASCII because it supports 8-bit character codes which can encode 256 characters.

UNICODE: Unicode encodes special texts from different languages, letters, symbols, etc. It can be said that ASCII is a subset of the Unicode encoding scheme. (8-bit, 16-bit, 32-bit)

COMPUTER MEMORY

- It is a storage area used to store data and instructions.

- It is of two types: primary memory and secondary memory

Primary Memory

It is made of circuits on silicon chip.

It holds instructions and data, when user program is under execution.

It is volatile in nature, contents will get released once user switch off computer (device)

Primary Memory

It is of two types – RAM (Random Access Memory) and ROM (Read Only Memory)

RAM – Now a days, computer uses different types of memory which can be organized in a hierarchy around the CPU. The memory at a higher level in the storage hierarchy has less capacity to store data, is more expensive and is fastest to access.

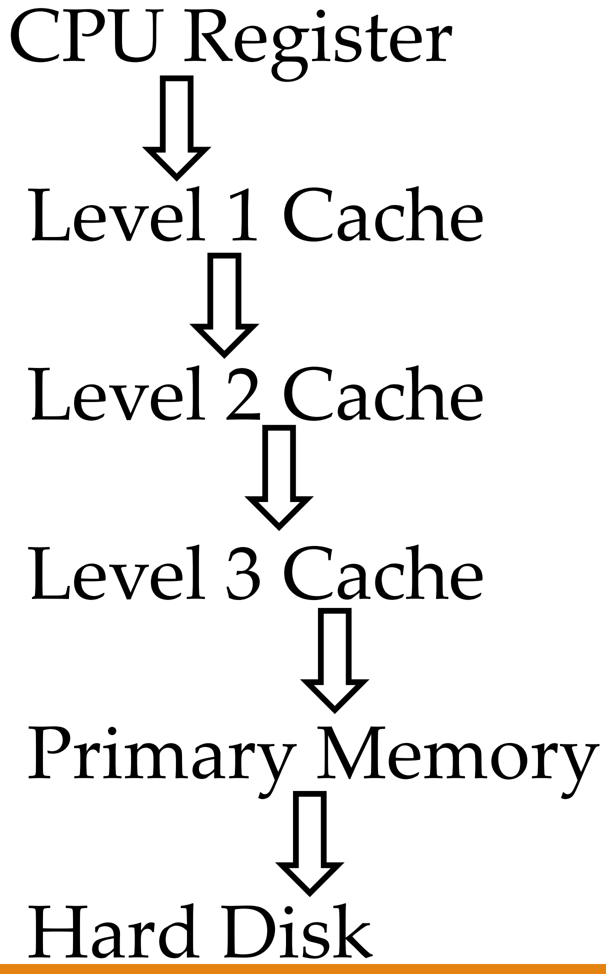
Primary Memory - RAM

Each register stores a word of data

Cache memory is an intermediate form of storage between ultra-fast registers and the RAM.

Dynamic RAM (DRAM), slower than Cache

Storage devices that are not directly accessible by the CPU



CPU Register

Level 1 Cache

Level 2 Cache

Level 3 Cache

Primary Memory

Hard Disk

Primary Memory - ROM

- ROM, means computer’s memory chips containing permanent or semi-permanent data.
- ROM is non-volatile. It is software directly written on hardware. Also called as firmware.
- ROM stores some critical programs such as BIOS which is used to boot the computer when it is turned on.
- Typical functions of ROM are to check input-output devices, to check RAM, to start booting process.
- Provided by manufacturer of computer.

Computer Software

- A program is a set of instructions that is arranged in a sequence to guide a computer to find a solution for the given problem.

- The process of writing a program is called programming.

- A software is a set of programs written using one or many computer languages.

Computer Software

Software is of two types –

1. System software

2. Application software

Computer Software – System S/W

- It is a collection of programs that enable the users to interact with hardware components efficiently
- It controls and manages the hardware
- System software is machine dependent
- The programmer must understand the architecture of the machine and hardware details to write a system software
- Interacts with the hardware directly
- Writing a system software is a complicated task
- Example: translators, compiler, operating system etc.

Computer Software – Application S/W

- It is a collection of programs written for a specific application. Like, we have library system, inventory control system, etc.
- It uses the services provided by the system software to interact with hardware components.
- It is machine independent.
- The programmer ignores the architecture of the machine and hardware details to write an application software.
- Interacts with the hardware indirectly through system calls provided by system software.
- Writing application programs is relatively very easy
- Example: MS-WORD, PAINT, Cricket08 etc.

Operating System

- Acts like a bridge between user and hardware
- Loads itself into RAM and then loads user application program as per need
- Enables user to execute programs
- Ensures that the system resources (such as CPU, memory, I/O devices, etc) are utilized efficiently

Operating System

Does –

- Memory management
- Process management
- I/O device management
- Secondary storage device management
- Provides protection (security)
- Provides user interface (CLI   /   GUI)

Translators

- It is a software which translate high level language program into machine understandable language file.

- It is machine specific.

- Compiler, Interpreter, Assembler are translator programs.

Interpreter

- It translates one instruction at a time into an intermediate form, which it then gets executes.

- The program execution is slow, because every time we execute the program, interpreter is used to translate the instructions.

- Debugging is tricky for big programs, since it reports error when a specific instruction which contains error is translated.

Compiler

- It translates entire program into machine language at a time.

- The program execution is fast, because once translated, compiler is no longer needed to execute the program.

- Debugging is fast, since it reports all the errors and translation is done only after program is bug free.

Linker

- In case of a large program, the programmers prefer to break a code into smaller modules.
- These modules are translated into object modules by compiler.
- Linker is a program that combines object modules to form an executable program. (.exe file)
- The compiler automatically invokes the linker as the last step in compiling a program.

C Programming

- C was developed in the early 1970s by Dennis Ritchie at Bell Laboratories.
- C was initially developed for writing system software.
- Now, C has become a popular language and various software programs are written using this language.
- Many other commonly used programming languages such as C++ and Java are also based on C.

Characteristics of C

- A high level programming language
- Small size. C has only 32 keywords. easy to learn.
- Function based language. Makes extensive use of function calls
- C is structured programming language.
- It supports loose typing (as a character can be treated as an integer and vice versa)
- Facilitates low level (bitwise) programming
- Supports pointers to refer computer memory, array, structures and functions.
- C is a portable language.

Uses of C

- C language is primarily used for native application development.
- C has been so widely accepted by professionals that compilers, libraries, and interpreters of other programming languages are often implemented in C.
- C is used as intermediate language for implementing other languages such as Node.js, Python, and Go.
- C is widely used to implement end-user applications

Structure of C program

- A C program contains one or more functions
- The statements in a C program are written in a logical sequence to perform a specific task.
- Execution of a C program begins at the main() function
- You can choose any name for the functions. Every program must contain one function that has its name as main().

Files used in a C Program

- Source file (.c file)
- Header file (.h file)
- Object Files (.o file)
- Binary Executable File (.exe file)

Using comments in a C Program

- It is a good programming practice to place some comments in the code to help the reader understand the code clearly.

- The compiler ignores the comments when forming the object file. This means that the comments are non-executable statements.

- // - single line comment   /* Write multi-line comment here */

KEYWORDS in a C Program

- C has a set of 32 reserved words often known as keywords. All keywords are basically words that have a fixed meaning. By convention all keywords must be written in lowercase (small) letters.
- Example: for, while, do-while, auto, break, case, char, continue, do, double, else, enum, extern, float, goto, if, int, long, register, return, short, signed, sizeof, static, struct, switch, typedef, union, unsigned, void, volatile

IDENTIFIERS in a C Program

- Identifiers are names given to program elements such as variables, arrays and functions.
- Rules for forming identifier name
  - it cannot include any special characters or punctuation marks (like #, $, ^, ?, ., etc) except the underscore"_".
  - There cannot be two successive underscores
  - Keywords cannot be used as identifiers
  - The names are case sensitive. So, example, “CODE” is different from “code” and “Code”.
  - It must begin with an alphabet or an underscore.
  - It can be of any reasonable length. Though it should not contain more than 31 characters.
- Example: roll_number, marks, name, emp_number, basic_pay, HRA, DA, dept_code, mark1, mark2

DATATYPES in a C Program

- A data type defines a domain of allowed values and the operations that can be performed on those values.
- There are four basic data types in C – int, char, float and double
- We can use type qualifiers with these basic data types
- There are two types of qualifiers –
  - Size qualifier – short and long
  - Sign qualifier – signed and unsigned (can be used with int and char only)

DATATYPES in a C Program

- The qualifier short and long can be applied to int type
- The qualifier long can be applied to double also to get long double which stores extended precision floating point number.

DATATYPES in a C Program

Name	**Description**	**Size***	**Range***
char	Character or small integer.	1byte signed:	-128 to 127 unsigned: 0 to 255
short int (short)	Short Integer.	2bytes	signed: -32768 to 32767 unsigned: 0 to 65535
int	Integer.	4bytes	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295
long int (long)	Long integer.	4bytes	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295
bool	Boolean value.	1byte	true or false
float	Floating point number.	4bytes	+/- 3.4e +/- 38 (~7 digits)
double	Double precision	8bytes	+/- 1.7e +/- 308 (~15 digits)
long double	Long double precision	8bytes	+/- 1.7e +/- 308 (~15 digits)

Note: Depends on compiler

DATATYPES in a C Program

Name	Description	Size*	Range*
char	Character or small integer.	1byte signed:	-128 to 127 unsigned: 0 to 255
short int (short)	Short Integer.	2bytes	signed: -32768 to 32767 unsigned: 0 to 65535
int	Integer.	4bytes	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295
long int (long)	Long integer.	4bytes	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295
bool	Boolean value.	1byte	true or false
float	Floating point number.	4bytes	+/- 3.4e +/- 38 (~7 digits)
double	Double precision	8bytes	+/- 1.7e +/- 308 (~15 digits)
long double	Long double precision	8bytes	+/- 1.7e +/- 308 (~15 digits)

Note: Depends on compiler

Escape Sequences in a C Program

Sequence	Meaning	ASCII Value
\a	Alert.	007
\b	Backspace.	008
\f	Form feed.	012
\n	Newline.	010
\r	Carriage return.	013
\t	Horizontal tab.	009
\v	Vertical tab.	011
\\	Backslash (\ ).	092
\′	Single quote (').	
\"	Double quote (").	
\?	Question mark (?).	

Conversion Specifications for I/O

The functions scanf() and printf() uses conversion specifications to specify the type and size of data.

Conversion character	Meaning
%c	single character
%d, %i	integer
%u	unsigned integer
%o	octal
%x	hexadecimal (a, b, c, d, e, f)
%X	hexadecimal (A, B, C, D, E, F)

Conversion Specifications for I/O

Conversion character	Meaning
%f	floating point number
%e	floating point number in exponential format
%E	
%s	string
%%	print % sign (not valid for scanf)
%p	print pointer in hexadecimal format (NVFscanf)

Operators in C

C language supports a lot of operators to be used in expressions. These operators can be categorized into the following major groups:

Arithmetic operators
Relational Operators
Equality Operators
Logical Operators
Unary Operators
Conditional Operators

Bitwise Operators
Assignment operators
Comma Operator
Sizeof Operator

TYPE CONVERSION AND TYPE CASTING

- Type conversion and type casting of variables refers to changing a variable of one data type into another.

- While type conversion is done implicitly, casting has to be done explicitly by the programmer.

TYPE CONVERSION AND TYPE CASTING

Type conversion is automatically done when we assign an integer value to a floating point variable. For ex,

```
float n;
        int m = 3;
        n = m;
```

Now, n = 3.0.

Type casting is also known as forced conversion. For ex,

```
 float salary = 10000.00;
int sal;            sal = (int) salary;
```

I/O in C Program

printf and scanf