Unit- I

Unit-1	
Computer	A computer is a device capable of performing computations and making logical decisions at a speed of millions, and even billions of times faster than human beings.
Hardware	Hardware is the equipment used to perform the necessary computations and includes the central processing unit (CPU), memories, monitor, keyboard, mouse, printer, and spike.
Software	Software consists of the programs that enable us to solve problems with computer by providing it with lists of instruction to perform.
Algorithm	A process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.
Flowchart	The pictorial representation of an algorithm using some shapes and symbols is known as flow chart.
Compiler	A program that converts instructions into a machine-code or lower-level form so that they can be read and executed by a computer.
Interpreter	A program that can analyse and execute a program line by line.
Datatype	A particular kind of data item, as defined by the values it can take, the programming language used, or the operations that can be performed on it.

Concepts:

1. Electronic Computers Then and Now (Computer Generations):-

Generation in computer terminology is a change in technology a computer is/was being used. Initially, the generation term was used to distinguish between varying hardware technologies. But nowadays, generation includes both hardware and software, which together make up an entire computer system. There are totally five computer generations known till date. Each generation has been discussed in detail

along with their time period, characteristics. We've used approximate dates against each generation which are normally accepted.

Following are the main five *generations* of computers

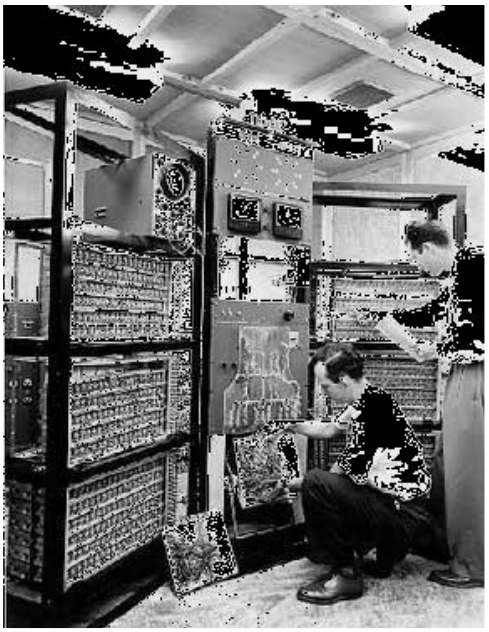
S.N.	Generation & Description
1	**First Generation** The period of first generation : 1942-1954. Vaccum tube based.
2	**Second Generation** The period of second generation : 1952-1964. Transistor based.
3	**Third Generation** The period of third generation : 1964-1972. Integrated Circuit based.
4	**Fourth Generation** The period of fourth generation : 1972-1990. VLSI microprocessor based.
5	**Fifth Generation** The period of fifth generation : 1990-onwards.ULSI microprocessor based

First Generation The period of first generation was 1942-1954. First generation of computer started with using vacuum tubes as the basic components for memory and circuitry for CPU(Central Processing Unit). These tubes like electric bulbs produced a lot of heat and were prone to frequent fusing of the installations, therefore, were very expensive and could be afforded only by very large organisations. In this generation mainly batch processing operating system were used. In this generation Punched cards, Paper tape, Magnetic tape Input & Output device were used.

There were Machine code and electric wired board languages used.The main features of First Generation are:

- Vacuum tube technology
- Unreliable

- Supported Machine language only
- Very costly
- Generate lot of heat
- Slow Input/output device
- Huge size
- Need of A.C.
- Non portable
- Consumed lot of electricity



Some computers of this generation were:

- ENIAC
- EDVAC
- UNIVAC
- IBM-701
- IBM-650

Second Generation The period of second generation was 1952-1964. This generation using the transistor was cheaper, consumed less power, more compact in size, more reliable and faster than the first generation machines made of vaccum tubes. In this generation, magnetic cores were used as primary memory and magnetic tape and magnetic disks as secondary storage devices. In this generation assembly language and high level programming language like FORTRAN, COBOL was used.

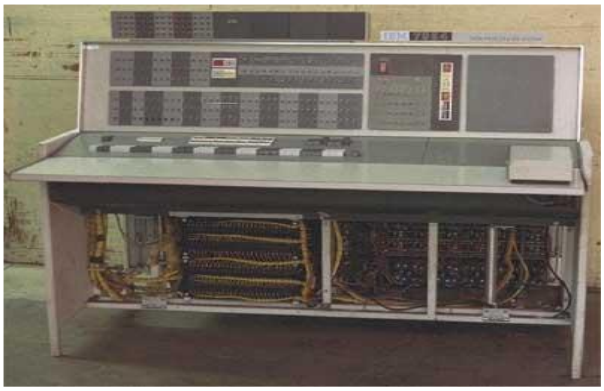
There was Batch processing and Multiprogramming Operating system used.

The main features of Second Generation are:

- Use of transistors
- Reliable as compared to First generation computers
- Smaller size as compared to First generation computers
- Generate less heat as compared to First generation computers
- Consumed less electricity as compared to First generation computers
- Faster than first generation computers
- Still very costly

- A.C. needed

Support machine and assembly languages Some computers of these generations are:



- IBM 1620
- IBM 7094
- CDC 1604
- CDC 3600
- UNIVAC 1108

Third Generation The period of third generation was 1964-1972. The third generation of computer is marked by the use of Integrated Circuits (IC's) in place of transistors. A single I.C has many transistors, resistors and capacitors along with the associated circuitry. The I.C was invented by Jack Kilby. This development made computers smaller in size, reliable and efficient. In this generation Remote processing, Time-sharing, Real-time, Multi-programming Operating System were used. High level language (FORTRAN-II TO IV, COBOL, PASCAL PL/1, BASIC, ALGOL-68 etc.) were used during this generation.

The main features of Third Generation are:

- IC used
- More reliable
- Smaller size
- Generate less heat
- Faster
- Lesser maintenance
- Still costly
- A.C needed
- Consumed lesser electricity
- Support high level language
- Some computers of this generation were:
- IBM-360 series
- Honeywell-6000 series
- PDP(Personal Data Processor)
- IBM-370/168
- TDC-316

Fourth Generation The period of Fourth Generation was 1972-1990. The fourth generation of computers is marked by the use of Very Large Scale Integrated (VLSI) circuits. VLSI circuits having about 5000 transistors and other circuit elements and their associated circuits on a single chip made it possible to have microcomputers of fourth generation. Fourth Generation computers became more powerful, compact, reliable, and affordable. As a result, it gave rise to personal computer (PC) revolution. In this generation Time sharing, Real time, Networks, Distributed Operating System were used. All the Higher level languages like C and C++, DBASE etc. were used in this generation.

The main features of Fourth Generation are:

- VLSI technology used
- Very cheap
- Portable and reliable
- Use of PC's
- Very small size
- Pipeline processing
- No A.C. needed
- Concept of internet was introduced
- Great developments in the fields of networks
- Some computers of this generation were:
- DEC 10
- STAR 1000
- PDP 11
- CRAY-1(Super Computer)
- CRAY-X-MP(Super Computer)

Fifth Generation The period of Fifth Generation is 1990-till date. In the fifth generation, the VLSI technology became ULSI (Ultra Large Scale Integration) technology, resulting in the production of microprocessor chips having ten million electronic components. This generation is based on parallel processing hardware and AI (Artificial Intelligence) software. AI is an emerging branch in computer science, which interprets means and method of making computers think like human beings. All the Higher level languages like C and C++, Java, .Net etc. are used in this generation. AI includes:

- Robotics
- Neural networks

- Game Playing
- Development of expert systems to make decisions in real life situations.
- Natural language understanding and generation.

The main features of Fifth Generation are:

- VLSI technology
- Development of true artificial intelligence
- Development of Natural language processing
- Advancement in Parallel Processing
- Advancement in Superconductor technology
- More user friendly interfaces with multimedia features
- Availability of very powerful and compact computers at cheaper rates Some computer types of this generation are:
- Desktop
- Laptop
- NoteBook
- UltraBook

Computer Hardware & Software:-

The elements of a computer system fall into two major categories:

Hardware: - Hardware is the equipment used to perform the necessary computations and includes the central processing unit (CPU), memories, monitor, keyboard, mouse, printer, and spike.

Software: - Software consists of the programs that enable us to solve problems with computer by providing it with lists of instruction to perform.

Computer Hardware of a computer consists of the following components:

- Main memory
- Secondary memory, which includes storage devices such as hard disks, CDs, DVDs, and flash drives.

- Central processing unit.
- Input devices, such as keyboards, mouse, touch pads, scanners, joysticks
- Output devices, such as monitors, printers, and speakers.

Memory Consists of memory cells, which is consist of ordered sequence of storage locations.

Address of a memory cell: the relative position of a memory cell in the computer's main memory.

Contents of a memory cell: The information stored in a memory cell, either a program instruction or data.

Bits and Bytes: Binary refers to a number system based on two numbers, 0 and 1, so a bit is either a 0 or a 1. Generally there are eight bits to a byte.

Storage and Retrieval of Information in Memory: A computer can either store a value into memory interims of binary number representations or retrieve a value from the memory

Main Memory: Main memory stores programs, data and results. Most computers have two types of main memory.

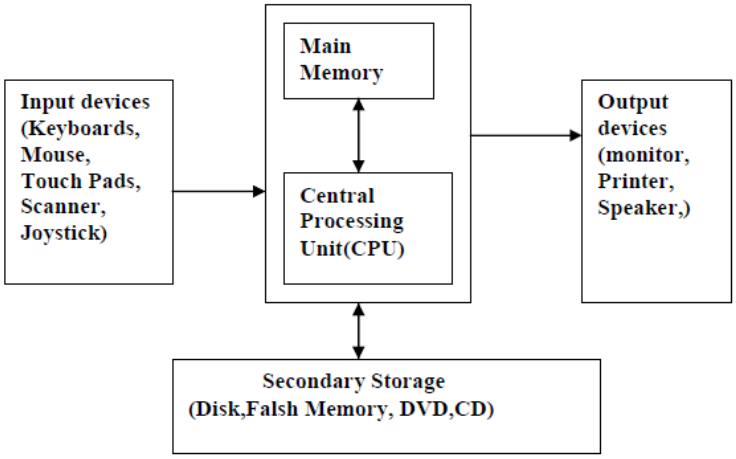
Random-Access-Memory (RAM) :Used for temporary storage of programs and data. Everything in RAM will be lost when the computer is switch is off.

Read-Only-Memory (ROM) : Used for permanently storage of programs and data. The computer can retrieve, but cannot store, information in ROM, hence its name, read-only.

Secondary Storage Devices: These devices are used to store huge amount of information. These are Hard disk, CDs, DVDs and flash memories.

Hard Disk: Hard disks are attached to their disk drives and coated with a magnetic material. Each data bit is a magnetized spot on the disc, and the spots are arranged in concentric circles called tracks. The disc drive read/write head accesses data by moving across the spinning disk to the correct track and then sensing the spots as they move by. Hard discs hold from one to several hundred gigabytes(GB) of data, but clusters of hard drives that stores data from an entire network may provide as much as a terabyte(TB) of storage.

Term	Abbreviation	Equivalent to	Comparison to Power of 10
Byte	B	8 Bits	
KiloByte	KB	1,024 (bytes)	>10**3
MegaByte	MB	1,048,576(bytes)	>10**6
GigaByte	GB	1,073,751,824(bytes)	>10**9
TeraByte	TB	1,099,511,627,776(bytes)	>10**12



Optical Drives: For storing and retrieving data on compact discs (CDs) or digital versatile disks (DVDs) that can be removed from the drive. One CD can hold 680 MB of data. A DVD uses smaller pits packed in a tighter spiral, allowing storage of 4.7 GB of data one layer. Some DVDs can hold four layers of data- two on each side for a total capacity of 17 GB, sufficient storage for as much as nine hours of studio quality video and multi-channel audio.

Flash Drive (USB)(Universal Serial Bus):Flash drives(stick memory) have no moving parts and all data transfer is by electronic signal only. Typical USB flash drives store 1 to a few GB of data, but 64 GB drives are also available. Information stored on a disk is organized into separate collections called files: One file may contain a C program. Another file may contain the data to be processed by the program (a data file) or a file may contain a picture information. A file could contain results generated by a program (an output file). The names of all files stored on a disk are listed in the disk's directory. This directory may be divided into one or more levels of subdirectories or folders.

Central Processing Unit The Central Processing Unit (CPU) has two roles.

A) Coordinating Operations of Computer

B) Performing Arithmetic and Logical operations on data.

The CPU follows the instructions contained in a computer program to determine which operations should be carried out and in what order. The CPU stores the results in main memory. The CPU can perform such arithmetic operations as addition, subtraction, multiplication, and division. The CPU can also compare the contents of two memory cells and make decision based on the result of that comparison. A CPU's current instruction and data values are stored temporarily inside the CPU in special high speed memory locations called registers (high speed memory location). **Input/output Devices** We use Input/output devices to communicate with the computer. They allow us to enter data for a computation and to observe the results

of that computation. We use a keyboard as an input device and monitor (display screen) as an output device. A keyboard has keys for letters, numbers, and punctuation marks plus some extra keys for performing special functions. A mouse is an input device used to select an operation. A monitor provides a temporary display of the information that appears on its screen. If you want a printed version (hard copy) of some information you must sent that information to an output device called printer.

Computer Software Mainly there are two main software groups in a computer. These are Operating system (OS) and Application software. To make the hardware friendly to the user operating system is used. The collection of computer programs that control interaction of user and computer hardware is called **Operating System**.

Application programs are developed to assist a computer user in accomplishing specific task. For example , a word-processing application MS Word, WordPerfect, or a spreadsheet application Lotus or Excel programs or a database management application such as Access or dBase systems are well known application programs.

STRUCTURE OF A C PROGRAM

Every C program is made of one or more preprocessor commands, a global declaration section and one or more functions.

1. The preprocessor commands are special instructions to the preprocessor that specifies how to prepare the program for compilation. One of the most important preprocessor commands we use commonly is include. They include command tells the preprocessor that we need information from selected libraries known as header files.

2. The global declaration section comes at the beginning of the program. The global variables

are declared here.

3. The execution of the program starts from the function called main(). All functions in a

program, including main, are divided into two sections one declaration section and the statement section.

a) Declaration section: it is at the beginning of the function. It describes the data that you will

be using in the function.

b) Statement Section: It contains the instructions to the computer that perform specific task.

First C Program

```
#include <stdio.h>
void main()
 {
printf(“ Hello! Welcome to C”);
}
```

Comments: used to make our code unreadable means the compiler ignores these comments when it translates the program into executable cod. C uses 2 different formats 1. Line commenting: it uses two slashes (//) to identify the comment and it is used to comment entire line. 2. Block Commenting: it uses two pair of tokens /* and */ to comment entire block of code.

Algorithm:-

Definition: A process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.

Fibonacci Series Algorithm:

1. Start
2. Declare variables i, a,b , show
3. Initialize the variables, a=0, b=1, and show =0
4. Enter the number of terms of Fibonacci series to be printed
5. Print First two terms of series
6. Use loop for the following steps

   show=a+b

   a=b

   b=show
7. Increase value of i each time by 1 print the value of show
8. End

Characteristics of an algorithm

1) Input: It may accept zero or more inputs

2) Definiteness: Each instruction must be well defined. There should not be any ambiguity.

3) Effectiveness: Each instruction must be simple and be carried out in a finite amount of time.

4) Finiteness: Algorithm should have finite sequence of instructions. It should not enter into an infinite loop.

5) Output: It must produce at least on output.

Advantages of algorithm:

Algorithm is written in English like language, logic of problem can be explained easily to others. It acts as blue print during a program development Helps in debugging so that we can identify the logical errors in a program. Program / Software maintenance becomes much easier.

Different patterns of algorithms:

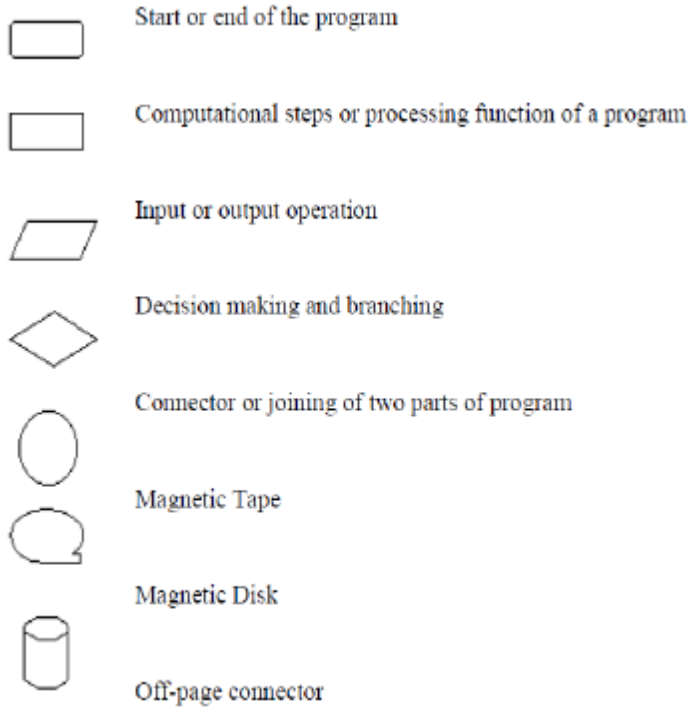
1) Sequential: In this different steps occur in a sequence

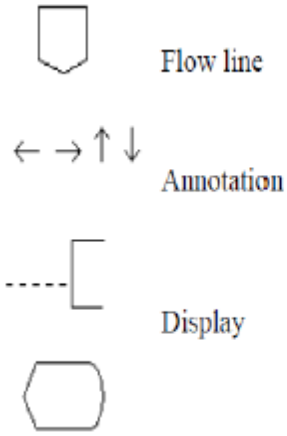
2) Conditional: In this different steps are executed based on a condition (whether true/false) . In programming languages, a conditional pattern is implemented using decision making statements.

3) Iteration: In this a task (one or more steps) is repeated more than once. In programming languages, an iteration pattern is implemented using loops.

Flowchart:-

The pictorial representation of an algorithm using some shapes and symbols is known as flow chart.





The Flowchart Symbols and Their Usage

- Terminal points indicate the starting [sometimes known as a 'trigger'] and ending points of a process.
- Step represents a single step within a process, and usually contains the name of a specific action.
- Page symbols refer to individual web pages, which may or may not contain multiple elements.
- File symbols represent those data elements that exist independently of navigational properties outside of that page, e.g., audio sounds, movie clips, or a portable document file (PDF).
- A decision point indicates a sequence in the process at which the end user chooses an option, i.e., a "yes-no", or "true-false" response, and then branches to different parts of the flowchart.
- Arrows and connecting lines diagram the logical progression through the course, subject to the choices made at decision or action points within the process.
- The input/action symbol represents a user response that directs the course flow from that point onwards, i.e., an online test, or questionnaire form. Represents the choice made by the user from mutually exclusive options, e.g., a student choosing among different lesson plans.
- Conditional selector is similar to the conditional branch except that the user has the option to choose from a number of paths that will fulfill the requested conditions, e.g., the results of a search engine request.

- Pages that share one or more common aspects, and are functionally identical may be simplified as a rounded corner rectangle, such as an on-line test or feedback form.
- Reference is used as a connecting point when the flowchart necessitates using more than one page, or refers to a complicated subroutine that would be impossible to contain on the main flowchart page.
- Annotations provide helpful comments or explanations, e.g. denoting the location where an undeveloped new page/process will fit into the navigational flow structure, or notes for specific team members for further development.
- Flow references and flow areas are symbols for reusable sequences, such as logging in with a specific user id and password to enter the course or to initiate an on-line quiz.
- The flow reference symbol acts as a place holder for the flow area sequence in the chart in every situation in which it is repeated.

Software Development Method

1. Specifying and analyzing the problem statement.
2. Designing an Algorithm
3. Coding and Implementation
4. Debugging
5. Testing and validating
6. Documentation and maintenance

Specifying and analyzing the problem statement: The problem which has to be implemented into a program must be thoroughly understood before the program is written. Problem must be analyzed to determine the input and output requirements of the program. A problem statement is created with these specifications.

Designing an Algorithm: With the problem statement obtained in the previous step, various methods available for obtaining the required solution are analyzed and the best method is designed into algorithm.

Coding and implementation: the actual problem is written in the required programming language with the help of information depicted in flow charts and algorithms.

Debugging: there is a possibility of occurrence of errors in programs. These errors must be removed to ensure proper working of programs. Hence solving the program without errors is known as debugging.

Types of errors that may occur in the program are:

a) Syntactic Errors (Compilation Errors): These errors occur due to the usage of wrong syntax for the statements.

b) Runtime Errors: These errors are determined at the execution time of the program.

c) Logical Errors: These errors occur due to incorrect usage of the instructions in the program.

Testing and Validating: Testing and Validation is performed to check whether the program is producing correct results or not for different values according to user requirement.

Documentation and Maintenance: Documentation is the process of collecting, organizing and maintaining, in written the complete information of the program for future references. Maintenance is the process of upgrading the program according to the changing requirements.

Basic Data Types

A type defines a set of values and set of operations that can be applied on those values. C language data types can be broadly classified as

1. Primary data type.
2. Derived data type
3. User-defined data type

Primary data type: All C Compilers accept the following fundamental data types.

1.	Integer	Int
2.	Character	char
3.	Floating Point	float
4.	Double Precision floating point	double
5.	Void	void

The size and range of each data type is given in the table below:

Data Type	Range Of Values
char	-128 to 127
int	-32768 to +32767
float	3.4 e -38 to 3.4 e +38
Double	1.7 e -308 to 1.7 e +308

Integer Type: Integers are whole numbers with a machine dependent range of values. C has 3 classes of integer storage namely short int, int and long int. All of these data types have signed and unsigned forms. A short int requires half the space than normal integer values. Unsigned numbers are always positive and consume all the bits for the magnitude of the number. The long and unsigned integers are used to declare a longer range of values.

Floating Point Types: Floating point number represents a real number with 6 digits precision. Floating point numbers are denoted by the keyword float. When the accuracy of the floating point number is insufficient, we can use the double to define the number. The double is same as float but with longer precision. To extend the precision further we can use long double which consumes 80 bits of memory space.

Character Type: A single character can be defined as a defined as a character type of data. Characters are usually stored in 8 bits of internal storage. The qualifier signed or unsigned can be explicitly applied to char. While unsigned characters have values between 0 and 255, signed characters have values from –128 to 127.

CONSTANTS

A constant value is the one which does not change during the execution of a program. C supports several types of constants.

1. Integer Constants
2. Real Constants
3. Single Character Constants
4. String Constants

Integer Constants:

An integer constant is a sequence of digits. There are 3 types of integer namely decimal integer, octal integers and hexadecimal integer.

Decimal Integers consists of a set of digits 0 to 9 preceded by an optional + or - sign. Spaces, commas and non digit characters are not permitted between digits.

Example for valid decimal integer constants are

123 -31 0 562321 + 78

Octal Integers constant consists of any combination of digits from 0 through 7 with a O at the beginning. Some examples of octal integers are

O26 O O347 O676

Hexadecimal integer constant is preceded by OX or Ox, they may contain alphabets from A to F or a to f. The alphabets A to F refers to 10 to 15 in decimal digits.

Example of valid hexadecimal integers are

OX2 OX8C OXbcd Ox

Real Constants: Real Constants consists of a fractional part in their representation. Integer constants are inadequate to represent quantities that vary continuously. These quantities are represented by numbers containing fractional parts like 26.082.

Example of real constants are

0.0026 -0.97 435.29 +487.0

Single Character Constants: A Single Character constant represent a single character which is enclosed in a pair of quotation symbols.

Example for character constants are ‘5’ ‘x’ ‘;’ ‘ ‘

All character constants have an equivalent integer value which is called ASCII Values.

String Constants: A string constant is a set of characters enclosed in double quotation marks. The characters in a string constant sequence may be a alphabet, number, special character and blank space. Example of string constants are

“VISHAL” ”1234” ”God Bless” ”!.....?”

Backslash Character Constants [Escape Sequences]: Backslash character constants are special characters used in output functions. Although they contain two characters they represent only one character. Given below is the table of escape sequence and their meanings.

Constant	**Meaning**
'\a'	Audible Alert (Bell)

'\b'	Backspace
'\f'	Formfeed
'\n'	New Line
'\r'	Carriage Return
'\t'	Horizontal tab
'\v'	.Vertical Tab
'\''	Single Quote
'\"'	Double Quote
'\?'	Question Mark
'\\'	Back Slash
'\0'	Null

OPERATORS

An operator is a symbol which helps the user to give instruction to the computer to do a certain mathematical or logical manipulations. Operators are used in C language program to operate on data and variables.

C has a rich set of operators which can be classified as follows:

1. Arithmetic operators
2. Relational Operators
3. Logical Operators
4. Assignment Operators
5. Increments and Decrement Operators
6. Conditional Operators
7. Bitwise Operators
8. Special Operators

1. Arithmetic Operators All the basic arithmetic operations can be carried out in C. All the operators have almost the same meaning as in other languages. Both unary and binary operations are available in C language. Unary operations operate on a singe operand, therefore the number 5 when operated by unary – will have the value –5.

Operator	Meaning

+	Addition or Unary plus
-	Subtraction or Unary Minus
*	Multiplication
/	Division
%	Modulus Operator

Examples of arithmetic operators are:

x + y , x - y , -x + y , a * b + c , -a * b etc.,

Here a, b, c, x, y are known as operands. The modulus operator is a special operator in C language which evaluates the remainder of the operands after division.

Integer Arithmetic When an arithmetic operation is performed on two whole numbers or integers than such an operation is called as integer arithmetic. It always gives an integer as the result. Let x = 27 and y = 5 be 2 integer numbers. Then the integer operation leads to the following results. x + y = 32 , x – y = 22 , x * y = 115 , x % y = 2 , x / y = 5 In integer division the fractional part is truncated.

Floating point arithmetic When an arithmetic operation is preformed on two real numbers or fraction numbers such an operation is called floating point arithmetic. The floating point results can be truncated according to the properties requirement. The remainder operator is not applicable for floating point arithmetic operands.

Let x = 14.0 and y = 4.0 then

x + y = 18.0 , x – y = 10.0 , x * y = 56.0 , x / y = 3.50

Mixed mode arithmetic When one of the operand is real and other is an integer and if the arithmetic operation is carried out on these 2 operands then it is called as mixed mode arithmetic. If anyone operand is of real type then the result will always be real thus 15/10.0 = 1.5

2. Relational Operators Often it is required to compare the relationship between operands and bring out a decision and program accordingly. This is when the relational operator come into picture. C supports the following relational operators.

Operator	**Meaning**
<	is less than
<=	Is less than or equal to
>	Is greater than
>=	Is greater than or equal to
==	Is equal to
!=	Is not equal to

It is required to compare the marks of 2 students, salary of 2 persons, we can compare them using relational operators.

A simple relational expression contains only one relational operator and takes the following form. exp1 relational operator exp2

Where exp1 and exp2 are expressions, which may be simple constants, variables or combination of them. Given below is a list of examples of relational expressions and evaluated values.

6.5 <= 25 TRUE , -65 > 0 FALSE , 10 < 7 + 5 TRUE

Relational expressions are used in decision making statements of C language such as if, while and for statements to decide the course of action of a running program.

3. Logical Operators

C has the following logical operators; they compare or evaluate logical and relational expressions.

Operator	Meaning
&&	Logical AND
||	Logical OR

!	Logical NOT

This operator is used to evaluate 2 conditions or expressions with relational operators simultaneously. If both the expressions to the left and to the right of the logical operator is true then the whole compound expression is true.

Example: a > b && x = = 10

The expression to the left is a > b and that on the right is x == 10 the whole expression is true only if both expressions are true i.e., if a is greater than b and x is equal to 10. Logical OR (||) The logical OR is used to combine 2 expressions or the condition evaluates to true if any one of the 2 expressions is true.

Example: a < m || a < n

The expression evaluates to true if any one of them is true or if both of them are true. It evaluates to true if a is less than either m or n and when a is less than both m and n. Logical NOT (!) The logical not operator takes single expression and evaluates to true if the expression is false and evaluates to false if the expression is true. In other words it just reverses the value of the expression.

For example

! (x >= y) the NOT expression evaluates to true only if the value of x is neither greater than or equal to y

4. Assignment Operators

The Assignment Operator evaluates an expression on the right of the expression and substitutes it to the value or variable on the left of the expression.

Example: x = a + b

Here the value of a + b is evaluated and substituted to the variable x. In addition, C has a set of shorthand assignment operators of the form.

var oper = exp;

Here var is a variable, exp is an expression and oper is a C binary arithmetic operator. The operator oper = is known as shorthand assignment operator

Example: x + = 1 is same as x = x + 1

The commonly used shorthand assignment operators are as follows

Simple Assignment	Shorthand Assignment
a=a+1	a+=1
a=a-1	a-=1
a=a*(n+1)	a*=(n+1)
a=a/(n+1)	a/=(n+1)
a=a%b	a%=b

5. Increment and Decrement Operators

The increment and decrement operators are one of the unary operators which are very useful in C language. They are extensively used in for and while loops. The syntax of the operators is given below

1. ++ variable name 2. variable name++

3. – –variable name 4. variable name– –

The increment operator ++ adds the value 1 to the current value of operand and the decrement operator – – subtracts the value 1 from the current value of operand. ++variable name and variable name++ mean the same thing when they form statements independently, they behave differently when they are used in expression on the right hand side of an assignment statement.

Consider the following

m = 5;

y = ++m; (prefix)

In this case the value of y and m would be 6 Suppose if we rewrite the above statement as

m = 5;

y = m++; (post fix)

Then the value of y will be 5 and that of m will be 6. A prefix operator first adds 1 to the operand and then the result is assigned to the variable on the left. On the other hand, a postfix operator first assigns the value to the variable on the left and then increments the operand.

6. Conditional or Ternary Operator

The conditional operator consists of 2 symbols the question mark (?) and the colon (:).

The syntax for a ternary operator is as follows:

exp1 ? exp2 : exp3

The ternary operator works as exp1 is evaluated first. If the expression is true then exp2 is evaluated & its value becomes the value of the expression. If exp1 is false, exp3 is evaluated and its value becomes the value of the expression. Note that only one of the expression is evaluated.

For example:

a = 10;

b = 15;

x = (a > b) ? a : b

Here x will be assigned to the value of b.

The condition follows that the expression is false therefore b is assigned to x.

Output

Input 2 integers : 34 45

The largest of two numbers is 45

7. Bitwise Operators

C has a distinction of supporting special operators known as bitwise operators for manipulation data at bit level. A bitwise operator operates on each bit of data. Those operators are used for testing, complementing or shifting bits to the right on left. Bitwise operators may not be applied to a float or double.

Operator	Meaning
&	Bitwise AND
|	Bitwise OR
^	Bitwise Exclusive OR
<<	Shift Left

>>	Shift Right

8. Special Operators

C supports some special operators of interest such as comma operator, size of operator, pointer operators (& and *) and member selection operators (. and ->).

The size of and the comma operators are discussed here.

The Comma Operator The comma operator can be used to link related expressions together. A comma linked list of expressions are evaluated left to right and value of right most expression is the value of the combined expression.

For example the statement

value = (x = 10, y = 5, x + y);

First assigns 10 to x and 5 to y and finally assigns 15 to value. Since comma has the lowest precedence in operators the parenthesis is necessary. Some examples of comma operator are

In for loops:

for (n=1, m=10, n <=m; n++,m++)

In while loops

While (c=getchar(), c != '10')

Exchanging values

t = x, x = y, y = t;

The size of Operator

The operator size of gives the size of the data type or variable in terms of bytes occupied in the memory. The operand may be a variable, a constant or a data type qualifier.

Example:

m = sizeof (sum);

n = sizeof (long int);

k = sizeof (235L);

The size of operator is normally used to determine the lengths of arrays and structures when their sizes are not known to the programmer. It is also used to allocate memory space dynamically to variables during the execution of the program.

<u>Assignment Questions</u>

Unit -I

1. What is secondary storage? Explain different secondary storage devices.
2. Draw and explain basic structure of computer.
3. Explain the general form of a C program with an example.
4. What is a constant? Explain the different types of constants in C.
5. List and explain the various symbols used in flowchart with figures.
6. What is an operator in C? What are the different operators available in C? Explain with suitable examples.
7. Explain the bitwise operators and relation operators available in C program.
8. Explain the categories of modern computers according to their size and performance.
9. Distinguish between High level language and Assembly language programming.
10. Explain different basic data types in C with examples.
11. Explain in detail about computer software.
12. What are the major computer hardware components? Explain them.
13. Explain in detail about the system development life cycle.