

**CLASS - XI**

# **COMPUTER SCIENCE**

SUPPORT MATERIAL 2023-24



**DIRECTORATE OF EDUCATION, GOVT. OF NCT,  
DELHI**

**LIST OF GROUP LEADER AND SUBJECT – EXPERTS  
FOR PREPARATION OF SUPPORT MATERIAL**

**CLASS – XI (2023-24)**

**SUBJECT: COMPUTER SCIENCE**

- 1. MR. TRILOK CHAND HALDUNIA** - **PRINCIPAL**  
**GSBV, TUGHLAKABAD EXTENSION**  
**SCHOOL ID: 1925056**
  
- 2. MR. PRASHANT KUMAR SINGH** - **LECTURER COMPUTER SCIENCE**  
**GSBV KIRAN VIHAR, DELHI - 92**  
**SCHOOL ID – 1001109**
  
- 3. MR. SACHIN KUMAR** - **LECTURER COMPUTER SCIENCE**  
**GSBV BURARI, DELHI – 84**  
**SCHOOL ID - 1207009**
  
- 4. MR. ANIRUDRA DIWAKAR** - **LECTURER COMPUTER SCIENCE**  
**GSBV SEEMAPURI, DELHI – 95**  
**SCHOOL ID - 1106009**
  
- 5. MS. YAMINI SHARMA** - **LECTURER COMPUTER SCIENCE**  
**SKB, H-BLOCK, PHASE-1**  
**ASHOK VIHAR**  
**SCHOOL ID - 11411026**
  
- 6. MR. AKSHAY KUMAR** - **LECTURER COMPUTER SCIENCE**  
**SV VASANT VIHAR**  
**SCHOOL ID - 1720024**
  
- 7. MR. AMIT BHARDWAJ** - **LECTURER COMPUTER SCIENCE**  
**KAUTILYA GSV(CO-ED)**  
**CHIRAG ENCL., SCHOOL ID - 1925003**

## TABLE OF CONTENTS

S.NO.	UNIT NAME	PAGE NO.
1	Computer Systems and Organisation	3 - 48
2	Computational Thinking and Programming -1	49 - 148
3	Society, Law, and Ethics	149 - 168
4	APPENDIX	169 - 177
5	SAMPLE PAPER	178 - 184

# Unit-1

## Computer Systems and Organization

### Basic Computer Organization

#### Introduction to Computer System

A computer is an **electronic device** that takes instructions and data from a user as **input**, **processes** them, and **produces** some meaningful **output** as a result. Computers are a **combination of hardware and software**. Computer systems work on the **IPO (Input-Process-Output)** Model. They process information stored as **binary digits (0s and 1s)** and perform complex tasks with incredible speed and accuracy.

- **Hardware are physical components** which can be **seen and touched (tangible)**. Examples of hardware include CPU, mouse, keyboard, monitor, motherboard, cables, CPU case, power supply unit, RAM, graphics card, sound card etc.
- Various hardware components are **interconnected together using a pathway called 'Bus'**, that facilitates the **transfer of data**.
- **Software is a set of instructions**, its documentation and data, which are **stored digitally** on the computer. Examples are Microsoft Windows, Linux, Paint, Word, PowerPoint, Photoshop, VLC Media Player, VS Code, Python. The software is intangible, i.e cannot be touched.
- A computer system receives instructions from the user (usually a human) using Input Devices.
- The Operating System (OS) is a crucial piece of software that allows users to interact with the computer hardware and other software easily.
- Computers process data through a series of instructions stored in their memory. The CPU fetches these instructions, decodes them, and then executes them.
- Computers can talk to other computers through networking. The Internet is one such network.

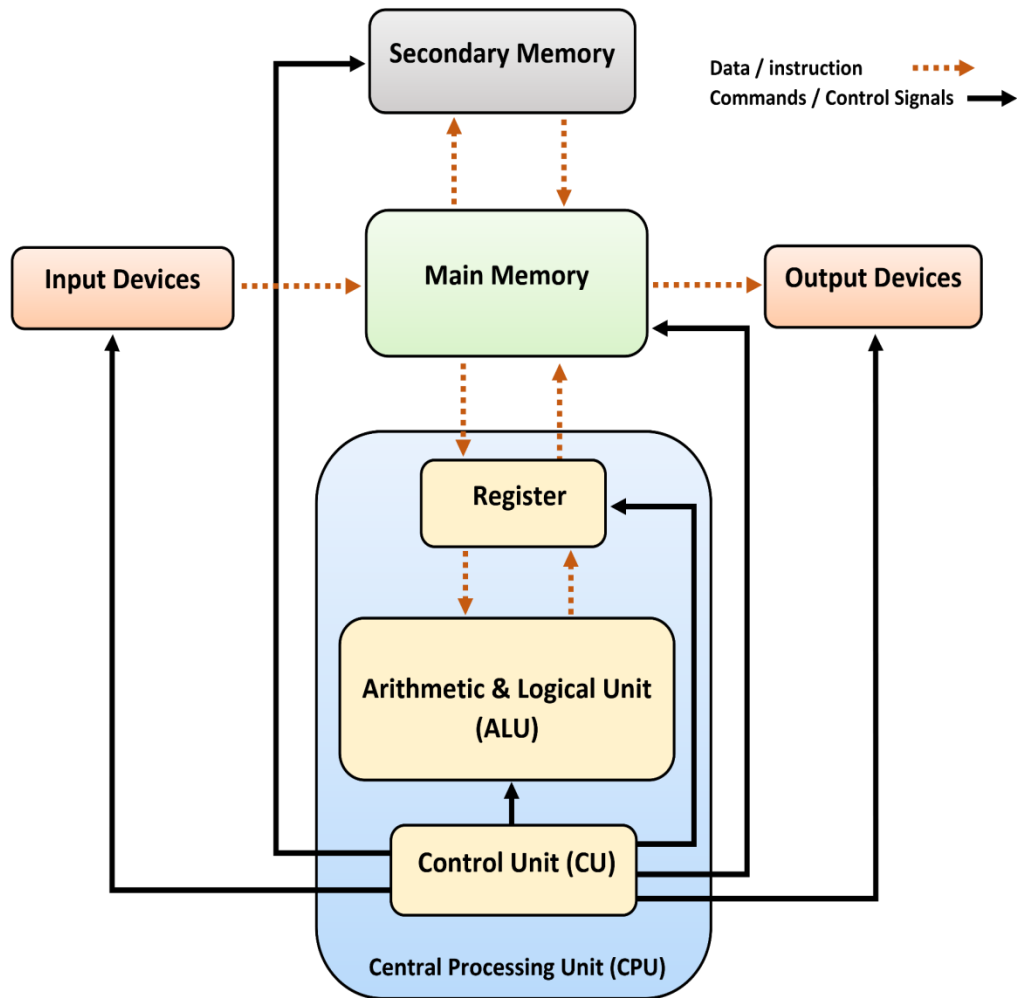
#### Advantages of Computers

- Multi-tasking - can perform multiple tasks simultaneously
- Speed - computes data rapidly
- Productivity - automates and streamlines work process to increase productivity
- Accuracy - tasks are performed with high accuracy
- Connectivity - access to vast amount of data and global communication networks
- Education- enhances learning through stores resources
- Storage - can store large amounts of data for a long time
- Reliability - can perform its functions adequately in a controlled environment with high precision

#### Disadvantages of Computers

- High initial costs of purchase, maintenance

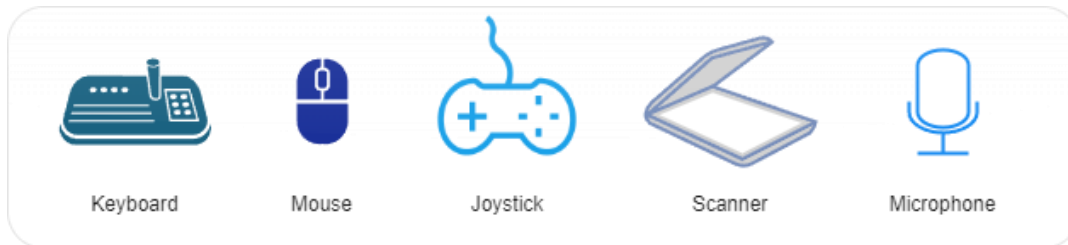
- Long use may lead to health issues
- Security Risks - vulnerable to hacking, loss and theft of information
- Environmental Impact due to high carbon emissions
- Software issues - problems may arise due to buggy software



**Computer Block Diagram**

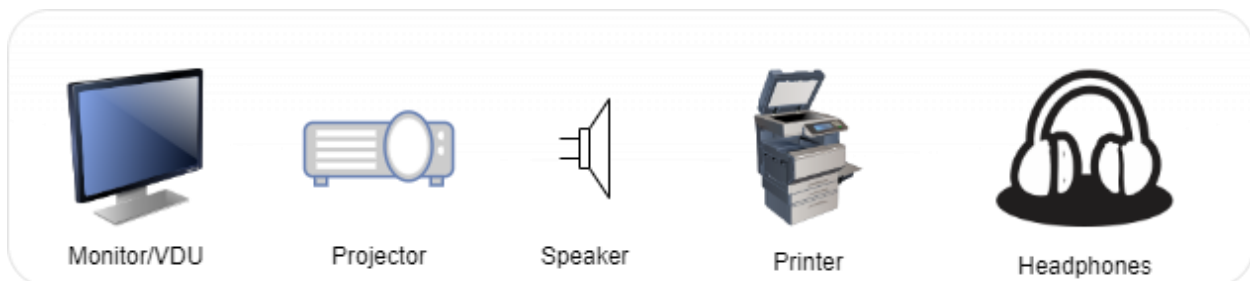
## Input Devices

- The hardware devices that **send input data from the user to the CPU** are known as Input Devices.
- They are responsible for capturing, and transferring data from the outside world into the computer system.
- Some of the popular Input Devices are Keyboard, Mouse, Joystick, Scanner, Microphone, Touchpad, Biometric Sensors, Graphic Tablet, Bar/QR Code Readers, Webcam, Magnetic Ink Character Reader (MICR) and Optical Character Reader (OCR) devices etc.



## Output Devices

1. The hardware devices that **are responsible for displaying output from the computer** are known as Output Devices.
2. Some of the popular Output Devices are Display Monitor, Projector, Speaker, Printer, Plotter, Headphones etc



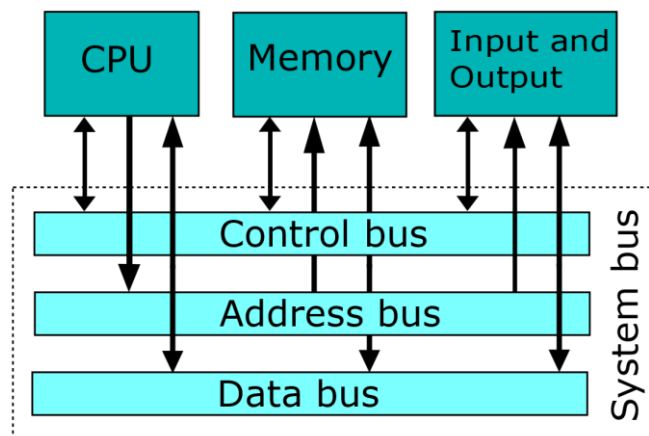
## Central Processing Unit

1. The **Central Processing Unit (CPU)** is the **brain** of the computer.
2. It is **responsible for processing** the instructions received, and generating the results.
3. The CPU contains **three major components, namely an Arithmetic and Logic Unit (ALU), a Control Unit (CU) and a set of Registers.**
4. The ALU performs arithmetic operations such as addition, subtraction, division, multiplication, exponentiation etc. It is also responsible for performing logical operations that result in either True or False.
5. The Control Unit (CU) is responsible for proper operation of the system. It controls the different operations of the computer by generating control signals.
6. The set of Registers are local/temporary storage for the CPU.

7. Registers can be accessed in a single CPU instruction and thus are extremely fast storage.
8. Registers are limited in number and thus data has to be transferred from the main/cache memory to the registers for processing of data.

### **Working of the CPU (Additional Reading)**

1. The CPU works on the concept of Information Processing Cycle (also known as Instruction Cycle).
2. Based on a clock signal, the CPU fetches instructions from the main memory.
3. To access the main memory, the CPU transmits signals from the System Bus.
4. The system bus is similar to a road network that connects various peripheral devices to the CPU. It is made up of Control Bus, Address Bus and the Data Bus. The Control Bus transmits controlling signals, the addresses are transmitted through the address bus, and the data is put onto the data bus.
5. The CPU then decodes the instruction into a sequence of operations.
6. It reads the effective address from the memory to fetch the data.
7. Then it performs the operation and stores the result back into the memory.



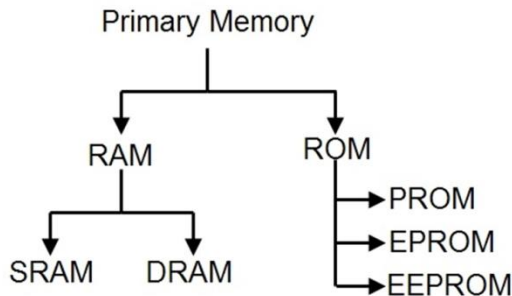
### **Memory**

1. The location where the computer stores data, instructions, and programs is called the Memory.
2. Computer memory is organized into memory cells, each of which stores a fixed amount of data, typically represented in binary as 0s and 1s.
3. The Memory of a computer is divided into Primary, Cache and Secondary Memory.

### **Primary Memory**

1. The primary memory is directly accessible by the CPU.
2. It is a high speed memory which can closely match the operational speed of the CPU.

3. Due to the high speed, the cost of primary memory is high and the storage capacity of primary memory is lower in comparison to the secondary memory.
4. There are two kinds of primary memory, namely RAM and ROM.



### Random Access Memory (RAM)

1. RAM, also known as the main memory, is a volatile memory, i.e the contents of the memory are lost as soon as the power to the system is switched off.
2. It is the main memory with which the CPU can interact directly.
3. It stores the instructions and data which are to be processed by the CPU shortly.
4. The data on the memory is stored in a fixed size block address.
5. The addresses on the RAM can be accessed in a random order, hence the name, Random Access.
6. RAM is majorly categorized into SRAM and DRAM.
7. DRAM stands for Dynamic RAM. Its contents have to be refreshed constantly. It is cheaper than SRAM.
8. SRAM stands for Static RAM. It does not require constant refreshing of the contents and hence is generally more expensive than DRAM.

### Read Only Memory (ROM)

1. ROM is a non-volatile memory.
2. The purpose of ROM is to store the software instructions required for starting up the computer.
3. The startup process is also called bootstrap or booting the computer.
4. It loads a software called Basic Input Output System (BIOS).
5. It is mainly of three types
  - a. PROM - Programmable ROM
  - b. EPROM - Erasable Programmable ROM
  - c. EEPROM - Electrically Erasable Programmable ROM

### Cache Memory

1. Cache memory is an intermediate memory between the CPU and the primary memory.
2. The need for cache memory arises from the difference in operational speed of the CPU and the primary memory.



- Often, the CPU is faster than the RAM/ main memory, and thus has to wait for the next instruction to be loaded from the memory.
- To reduce this waiting time, and increase the throughput of the CPU, a cache memory (buffer) is placed between the CPU and primary memory.
- The cache holds frequently used instructions and data and makes them readily available to the CPU.

## Secondary Memory

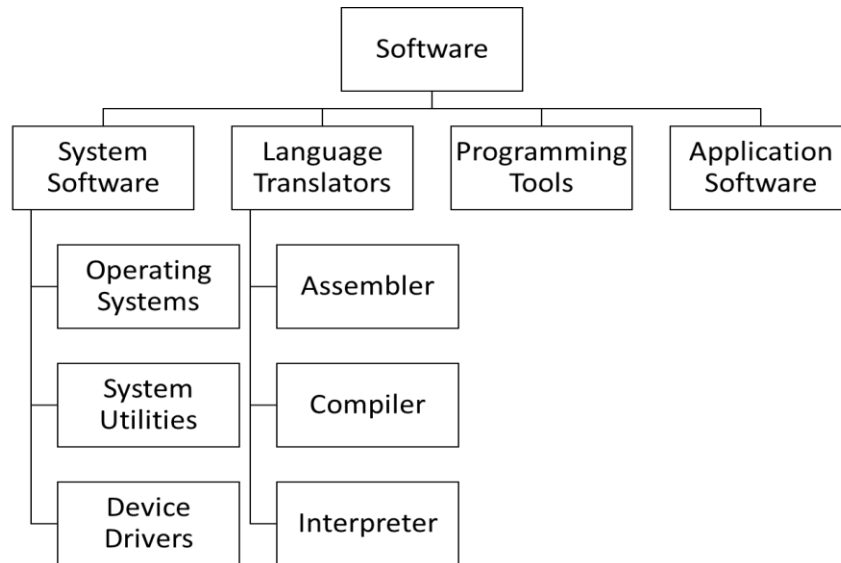
- Secondary memory is also known as **permanent** memory.
- It is **non-volatile** in nature, i.e the **data does not get lost even after power is turned off** from the computer.
- Examples of Secondary memory are Hard Disk Drive, Solid State Drive, USB Flash Drive.
- The CPU does not interact with the Secondary Memory directly.

## Units of Memory

- The smallest unit of memory is called a binary digit, or a bit.
- The bit can assume only two values, 0 and 1.
- A sequence of 4-bits is called a Nibble.
- A Word is a fixed length sequence of bits which the processor can handle at a time.

Sr. No	Unit	Remarks
1	Bit (b)	Can be 0 or 1
2	Nibble	1 nibble = 4 bits
3	Word	Group of bits on which the CPU can work as a single unit. Can be 8 bits, 16 bits, 32 bits or 64 bits depending on CPU Architecture.
4	Byte (B)	1 B = 8 bits
5	KiloByte(KB)	1 KB = 1024 B
6	MegaByte(MB)	1 MB = 1024 KB
7	GigaByte(GB)	1 GB = 1024 MB
8	TeraByte(TB)	1 TB = 1024 GB
9	PetaByte(PB)	1 PB = 1024 TB

## Types of Software



### System Software

It is software which controls all the operations of the computer system and interacts with the hardware connected to the computer. The computer cannot operate without system software. System software is majorly categorized into: Operating System, System Utilities and Device Drivers.

### Operating System

An operating system is system software that acts as an interface between the user and the computer hardware and manages all the resources of a computer. Examples of operating systems are Microsoft Windows, Ubuntu, Android, Apple iOS etc.

### System Utilities

The software that performs maintenance and configuration of the computer system is called System Utility.

Examples include Antivirus software, Disk Defragmentation Tool, System Restore Utility, Disk Partitioning Utility etc.

### Device Drivers

The software that directly interacts with a particular hardware or peripheral device is called the device driver. Each hardware has its unique device driver without which the operating system cannot communicate with the device. Examples include RealTek Audio Driver, NVidia Video Driver, MS USB Driver etc.

## Language Translators

Languages are majorly divided into two categories: **Low Level Languages and High-Level Languages.**

- Low Level Languages are nearer to Machine Code than to human-like languages. They are difficult to understand by humans but easily understandable by machines.
- On the other hand, humans can easily understand high level languages but computers require language translators to convert high level languages into low level languages which they understand.
- Low level language comprises Binary Language (combination of 0's and 1's) and Assembly Language (uses keywords like ADD, SUB, STR), whereas High Level Language comprises C, C++, Java, Python etc.
- Software that translates one language to another language is called language translators.
- Language Translators can be divided into Compiler, Interpreter and Assembler.

## Compiler, Interpreter and Assembler

- A compiler is a language translator that converts high level language into low level language at once. It shows all errors together with line number. Once all errors are corrected and object code is created, compiler is no more required in memory.
- An Interpreter is a language translator that converts high level language into low level language line by line, instead of converting the entire code at once. It stops at the line where error is found and requires rectification of same to move forward. It is always required in memory.
- Assembly language code can be converted to machine code (Binary Code) using translator called Assembler.

## Programming Tools

Tools that assist the users/developers in creating, editing, testing and debugging related to the development of software are known as Programming Tools.

They support completion of tasks during the development phase. It includes development tools, code editors and translators.

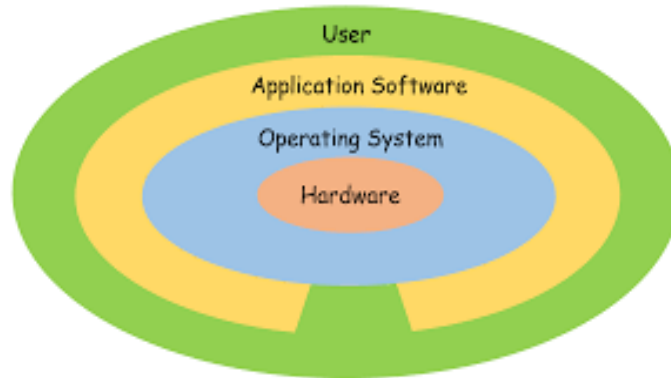
Examples include : Visual Studio Code, Sublime Text, Eclipse etc.

## Operating System

An operating system is system software that acts as an interface between the user and the computer hardware and manages all the resources of a computer.

## Need of Operating System

The main goal of an operating system is to provide a user-friendly environment, to use the available resources in an optimal and efficient manner and to provide services for building and running applications.



## Examples of Operating System

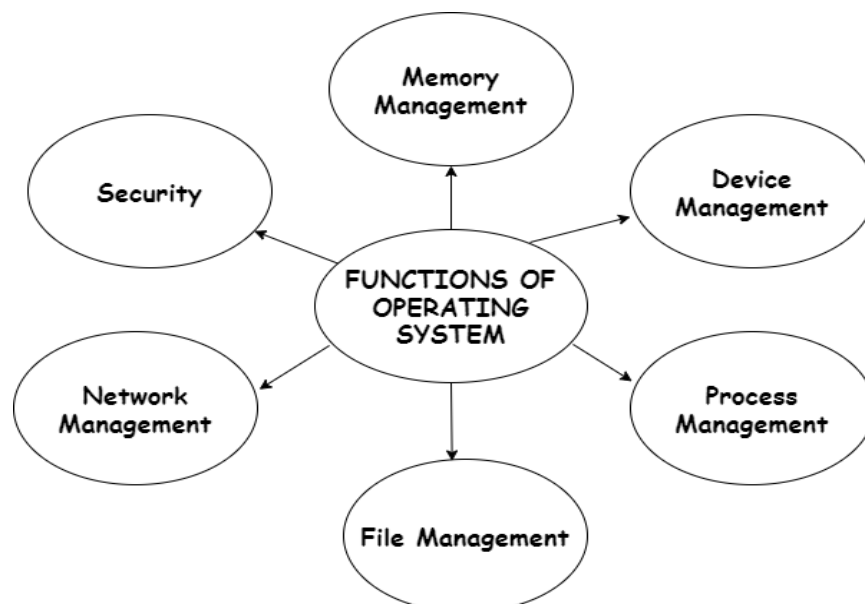
### 1. Desktop Operating System

- a. Microsoft Windows
- b. Apple's macOS
- c. Ubuntu (Linux based open source OS)
- d. Google's ChromeOS

### 2. Mobile Operating System

- a. Google Android
- b. Apple iOS
- c. Symbian
- d. Microsoft Windows RT

## Functions of Operating System



## Memory Management

The operating system manages the memory resources of a computer system. It keeps a record of used and available memory. It includes:

- Memory Allocation, Reallocation and Deallocation
- Memory Mapping
- Memory Swapping

## Device Management

Operating system manages communication among all the devices, keeps track of all the devices, allocates and deallocates devices. It includes:

- Device initialization
- Device Configuration
- Device Scheduling

## Process Management

It manages all the programs in execution (processes). It includes:

- Process Creation & Termination
- Process Scheduling
- Context Switching

## File Management

The operating system manages all the file management tasks. It keeps track of location, status, storage and operations of a file. It includes:

- File Creation and Deletion
- File Permissions and Security
- File Compression and Encryption
- File Sharing

## Network Management

It provides services to organize and maintain the network. It allows computers in a network to communicate with each other. It includes:

- Network Configuration
- Network Connectivity
- Network Resource Management

## Security

The operating system protects the system from all the threats, unauthorized access and other vulnerabilities. It includes:

- Authentication
- Authorization
- Data Encryption
- Firewall Management

## Other tasks

Some other tasks performed by OS are Job Accounting, Error Detection, Control over System Performance, Resource Allocation, Information and Resource Protection, and Handling I/O Operations.

## OS User Interface

Operating system acts as an interface between the user and the hardware. There are types of interfaces to perform different tasks on the basis of requirement.

### Command Line Interface

It is the interface where the user interacts with the system through commands. All the operations are carried out on the basis of commands entered by the user.

### Graphical Based User Interface

It is the interface that allows users to interact with the system through graphical icons, menu, taskbar etc.

### Gesture Based User Interface

It is the interface that uses physical gestures to operate the computer system. It allows users to perform tasks without physically touching the system.

### Voice Based User Interface

It is the interface that allows users to interact with a computer through voice commands. Examples of Voice User Interfaces are Google Assistant, Siri and Alexa.

### Touch Based User Interface

Touch based user interface requires a physical touch through the input device.

---

## Boolean Logic

Boolean Logic is a concept that involves binary variables and operations. It focuses on the values true and false (1 and 0). It was developed by the English Mathematician and logician George Boole. Boolean Algebra comprises of following:

1. Boolean Expression
2. Boolean Variable

### **Boolean Variable**

A boolean variable is a variable that holds boolean values True/ False or 1/0.

### **Boolean Expression**

A boolean expression is an expression that consists of a combination of boolean variables, boolean values and boolean operators. A boolean expression evaluates to either True or False.

### **Boolean Operators**

Boolean operators perform operations on operands (Boolean Variables/Values). The boolean operators are: AND, OR and NOT.

AND operator - It evaluates to True(1) if all inputs are True(1), otherwise False(0). It is represented by the dot operator (.)

Example : A.B may be read A AND B

OR operator - It evaluates to True(1) if any of the inputs is True(1), otherwise False(0). It is represented by the + operator.

Example: A + B may be read as A OR B.

NOT operator - It evaluates to True(1) when the condition is False, and returns False(0) when the condition is True(1).

Example: A' may be read as A complement.

$\bar{A}$  may be read as A complement.

### **Truth Table**

A truth table is a representation of all possible combinations of the input variables and the corresponding output values.

The number of rows in a truth table are  $2^n$ , where n is no. of input variables.

Example- Here, A and B are the input boolean variables, OR (+) is the operator, F is the output.

A	B	F = A OR B
0	0	0
0	1	1
1	0	1
1	1	1

### Rules of Boolean Logic (Additional Reading)

Name of Rule	AND Version	OR Version
Identity Law	$1.A=A$	$0+A=A$
Null Law or Dominant Law	$0.A=0$	$1+A=1$
Idempotent Law	$A.A=A$	$A+A=A$
Inverse Law	$A.A'=0$	$A+A'=1$
Commutative Law	$AB=BA$	$A+B=B+A$
Associative Law	$(AB)C=A(BC)$	$(A+B)+C=A+(B+C)$
Distributive Law	$A+BC=(A+B)(A+C)$	$A(B+C)=AB+AC$
Absorption Law	$A(A+B)=A$	$A+AB=A$
De Morgan's Law	$(AB)'=A'+B'$	$(A+B)'=A'.B'$

### De Morgan's Laws

DeMorgan's First Law states that when two input variables are AND'ed and complemented, they are equivalent to the OR of the complements of the individual variables.

$$(AB)'=A'+B'$$

A	B	A'	B'	AB	(AB)'	A'+B'
0	0	1	1	0	1	1
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	1	0	0



DeMorgan's Second Law states that when two input variables are OR'ed and complemented, they are equivalent to the AND of the complements of the individual variables.

$$(A+B)' = A'.B'$$

A	B	A'	B'	A+B	(A+B)'	A'.B'
0	0	1	1	0	1	1
0	1	1	0	1	0	0
1	0	0	1	1	0	0
1	1	0	0	1	0	0

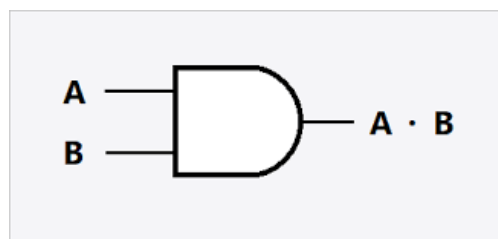
## Logic Gates

A logic gate is a device that performs a Boolean Function. One or more inputs in the form of 1/0 are provided to get the specific output governed by a logic. Examples of Logic Gates are: AND, OR, NOT, NOR, NAND and XOR.

NAND and NOR are called Universal Gates, as they can implement any Boolean expression.

### AND Gate

Logic Gate Diagram

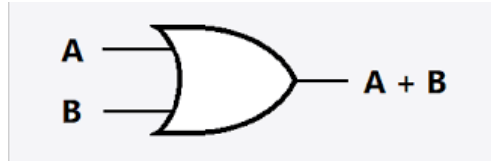


### Truth Table

A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

## OR Gate

Logic Gate Diagram

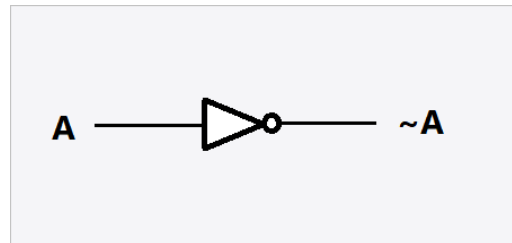


## Truth Table

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

## NOT Gate

Logic Gate Diagram



## Truth Table

A	A'
0	1
1	0

## NAND Gate

Logic Gate Diagram

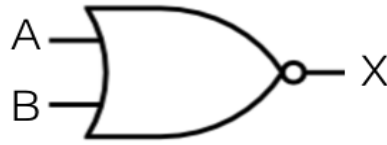


## Truth Table

A	B	A.B	(A.B)'
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

## NOR Gate

Logic Gate Diagram

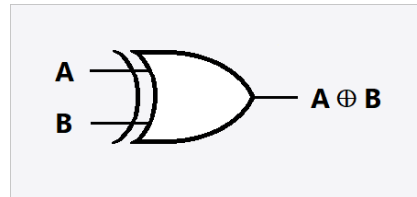


A	B	$A+B$	$(A+B)'$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

## Truth Table

## XOR Gate

Logic Gate Diagram



## Truth Table

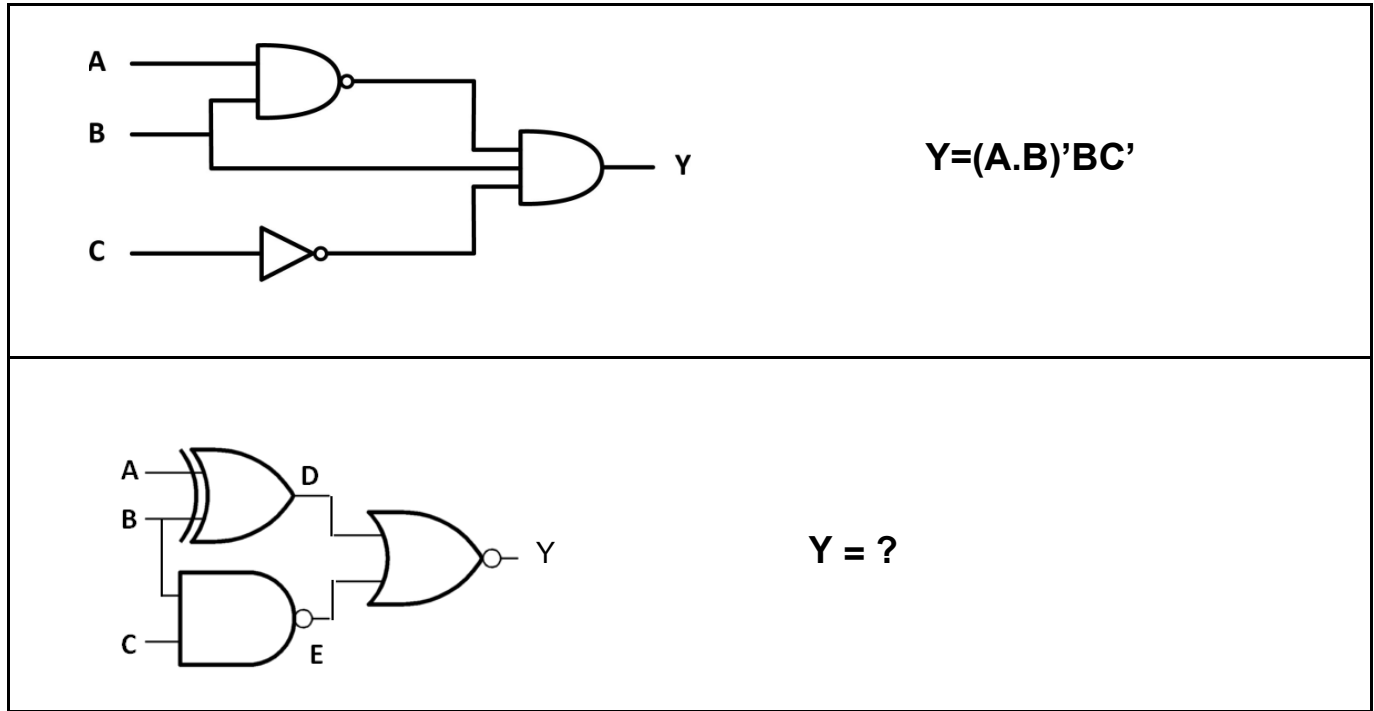
A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

XOR is represented as  $AB' + A'B$

## Logic Circuits

A logic circuit is an electronic circuit which performs logical operations on the input boolean variables, and transforms them into the output using a combination of Logic Gates.

Examples of a circuit diagram are:



## Number System

There are many possible ways to represent numbers. A number system provides a consistent and unique method to represent the numbers.

## Positional Number System

A positional number system is one way of writing numbers. It has unique symbols for 0 through  $(b - 1)$ , where  $b$  is the base (also known as radix) of the system. These symbols are called digits.

Some popular positional number systems are:

1. Binary - base 2
2. Octal - base 8
3. Decimal - base 10
4. Hexadecimal - base 16

## Decimal Number System

A decimal number uses base-10 for representing numbers. The numbers 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9 are the digits present in the system. Every number can be represented as a sum of powers of 10, with each power weighted by a digit.

For example, in the decimal system, 534 can be represented as

$$534 = 5 \times 10^2 + 3 \times 10^1 + 4 \times 10^0$$

Fractional numbers can also be represented using similar notation, for example, 123.238 can be represented as

$$123.238 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 2 \times 10^{-1} + 3 \times 10^{-2} + 8 \times 10^{-3}$$

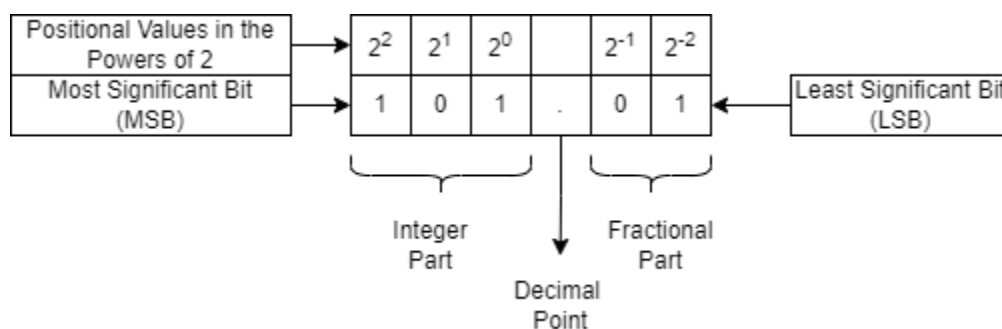
The powers of 10, which correspond respectively to the digits in a decimal integer when read from right to left, are called the place values of the digits.

## Binary Number System

The binary number system uses only two symbols, 0 and 1. Each numeral is known as a binary digit or a bit. It is also known as the base-2 numeral system, where the positional digits are powers of 2.

The binary system is most easily implemented through computer hardware. The binary digits 0 and 1 correspond to the presence or the absence of a digital signal. The manipulation of binary digits is performed using a combination of boolean logic gates and circuits. Negative numbers can also be represented in the Binary Number System.

The numbers are represented by creating a sequence of bits, such as 00000, 010100, 011100 etc. Binary numerals are often subscripted with 2 in order to indicate the base. For example,  $(110101)_2$ . Numbers that have no fractional part are called binary integers.



In the figure above, representation of the binary number  $(101.01)_2$  is shown. The most significant bit (MSB) is the bit in a binary number sequence with the largest value. This is usually the bit farthest to the left. The MSB represents the highest-order place of the binary integer. On the other hand, the Least Significant Bit (LSB) is the right-most bit of the number.

The maximum possible sequence of binary numbers that can be created using a sequence of n-bits is  $2^n$ .

No. of Bits	Count of possible binary numbers
1	$2^1 = 2$
2	$2^2 = 4$
3	$2^3 = 8$
4	$2^4 = 16$

## Octal Number System

Octal system, or the base-8 system uses 8 unique symbols, 0, 1, 2, 3, 4, 5, 6, and 7. The place values in the octal system are powers of 8. We can use 3-bit binary numbers to represent the octal digits, since  $2^3 = 8$ .

Decimal Number	3-bit Binary Number	Octal Number
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7

For example, an octal number  $(532.67)_8$  can be represented in the decimal number system as

$$(532.67)_8 = 5 \times 8^2 + 3 \times 8^1 + 2 \times 8^0 + 6 \times 8^{-1} + 7 \times 8^{-2}$$

Octal numbers find applications where the number of bits in one word is a multiple of 3. They are also used as shorthand for representing file permissions on UNIX/Linux operating systems and representation of UTF8 numbers, etc. The advantage of using Octal numbers is that it uses fewer digits than the decimal number system. It has fewer computations and is less prone to the computational errors.

## Hexadecimal Number System

The hexadecimal number system uses 16 unique symbols, 0-9 and A-F for representation of numbers. The place values in the hexadecimal system are powers of 16. Each hexadecimal digit has a unique 4-bit representation since  $2^4 = 16$ , so all possible digits can be represented with 4 bit binary numbers. The hexadecimal numbering system is often used by programmers to simplify the binary numbering system since large binary numbers can be organized and written neatly in their respective hexadecimal notations.

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

Some popular uses of Hexadecimal numbers are:

- Representation of addresses in memory.
- Representation of colors on web pages.
- Representation of Media Access Control (MAC) addresses.
- Display error messages.

## Number System Conversions

### 1) Decimal Number System to Other Base

- a) Decimal Number System to Binary Number System
- b) Decimal Number System to Octal Number System
- c) Decimal Number System to Hexadecimal Number System

### 2) Other Base to Decimal Number System

- a) Binary Number System to Decimal Number System
- b) Octal Number System to Decimal Number System
- c) Hexadecimal Number System to Decimal Number System

### 3) One Base to Other Base System

- a) Binary Number System to Hexadecimal Number System
- b) Hexadecimal Number System to Binary Number System
- c) Binary Number System to Octal Number System
- d) Octal Number System to Binary Number System
- e) Hexadecimal Number System to Octal Number System
- f) Octal Number System to Hexadecimal Number System

## Decimal Number System to Binary Number System Conversion

### Steps for the Integer part

1. For the integral part of the number, repeatedly divide the number by 2, and save the remainder.
2. Repeat the process until no further division is possible.
3. Read the remainders from the bottom to top to get the binary number.

### Steps for the fractional part

1. Repeatedly multiply the fractional part by 2 and note the integer and fractional part of the product.
2. Repeatedly multiply the fraction part by 2 and note the integer part from top to the bottom. (Stop if you get 0)
3. Repeat this procedure till sufficient precision is reached (usually 2 to 3 places after the radix point).



**Ques 1 - Convert decimal number 75 to binary number.**

Divisor	Dividend	Remainder	
2	75		
2	37	1	
2	18	1	
2	9	0	
2	4	1	
2	2	0	
2	1	0	
	0	1	

$= (1001011)_2$

**Ques 2- Convert decimal number 142.67 to a binary number.**

Divisor	Dividend	Remainder	
2	142		
2	71	0	
2	35	1	
2	17	1	
2	8	1	
2	4	0	
2	2	0	
2	1	0	
	0	1	

$.67 \times 2 = 1.34$	1	.34
$.34 \times 2 = 0.68$	0	.68
$.68 \times 2 = 1.36$	1	.36
$.36 \times 2 = 0.72$	0	.72



$= (10001110.1010)_2$

## Decimal Number System to Octal Number System Conversion

### Steps for the Integer part

1. For the integral part of the number, repeatedly divide the number by 8, and save the remainder.
2. Repeat the process until no further division is possible.
3. Read the remainders from the bottom to top to get the binary number.

### Steps for the fractional part

1. Repeatedly multiply the fractional part by 8 and note the integer and fractional part of the product.
2. Repeatedly multiply the fraction part by 8 and note the integer part from top to the bottom. (Stop if you get 0)
3. Repeat this procedure till sufficient precision is reached (usually 2 to 3 places after the radix point).

**Ques 1 - Convert decimal number 104 to octal number.**

Divisor	Dividend	Remainder	
8	104		
8	13	0	↑ LSB
8	1	5	
	0	1	

= (150)<sub>8</sub>

**Ques 2 - Convert decimal number 563.93 to octal number.**

Divisor	Dividend	Remainder	
8	563		
8	70	3	↑ LSB
8	8	6	
8	1	0	
	0	1	

.93 x 8 = 7.44	↓	7	.44
.44 x 8 = 0.88		0	.88
.88 x 8 = 7.04		7	.04
.04 x 8 = 0.32		0	.32

= (1063.7070)<sub>8</sub>

## Decimal Number System to Hexadecimal Number System Conversion

### Steps for the Integer part

1. For the integral part of the number, repeatedly divide the number by 16, and save the remainder.
2. Repeat the process until no further division is possible.
3. Read the remainders from the bottom to top to get the binary number.

### Steps for the fractional part

1. Repeatedly multiply the fractional part by 16 and note the integer and fractional part of the product.
2. Repeatedly multiply the fraction part by 16 and note the integer part. (Stop if you get 0)

3. Repeat this procedure till sufficient precision is reached (usually 2 to 3 places after the radix point).

**Ques 1 - Convert decimal number 600 to hexadecimal number.**

Divisor	Dividend	Remainder		
16	600			
16	37	8	↑	LSB
16	2	5		
	0	2		
= (258) <sub>16</sub>				

**Ques 2 - Convert decimal number 49 to hexadecimal number.**

Divisor	Dividend	Remainder		
16	49			
16	3	1	↑	LSB
	0	3		
= (31) <sub>16</sub>				

## Binary Number System to Decimal Number System Conversion

### Steps:

1. Multiply each digit with its respective weighted power of 2.
2. Calculate its sum.

**Ques 1 - Convert Binary Number 101101 into Decimal Number.**

$$\begin{aligned}
 &101101 \\
 &= 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\
 &= 32 + 0 + 8 + 4 + 0 + 1 \\
 &= 45
 \end{aligned}$$

**Ques 2 - Convert Binary Number 110101.010 into Decimal Number.**

$$\begin{aligned}
 &110101.010 \\
 &= 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} \\
 &= 32 + 16 + 0 + 4 + 0 + 1 + 0 + 0.25 + 0 \\
 &= 53.25
 \end{aligned}$$

## Octal Number System to Decimal Number System Conversion

### Steps:

1. Multiply each digit with its respective weighted power of 8.
2. Calculate its sum.

**Ques 1 - Convert Octal Number 156 into Decimal Number.**

$$\begin{aligned} &156 \\ &= 1 \times 8^2 + 5 \times 8^1 + 6 \times 8^0 \\ &= 64 + 40 + 6 \\ &= 110 \end{aligned}$$

**Ques 2 - Convert Octal Number 78.12172 into Decimal Number.**

$$\begin{aligned} &78.12 \\ &= 7 \times 8^1 + 8 \times 8^0 + 1 \times 8^{-1} + 2 \times 8^{-2} \\ &= 56 + 8 + 0.125 + 0.03125 \\ &= 64.15625 \end{aligned}$$

## Hexadecimal Number System to Decimal Number System Conversion

### Steps:

1. Multiply each digit with its respective weighted power of 16.
2. Calculate its sum.

**Ques 1 - Convert Hexadecimal Number 0.87 into Decimal Number.**

$$\begin{aligned} &0.87 \\ &= 0 \times 16^0 + 8 \times 16^{-1} + 7 \times 16^{-2} \\ &= 0 + 0.5 + 0.02734375 \\ &= 0.52734375 \end{aligned}$$

## Binary Number System to Hexadecimal Number System

There are two ways to convert a binary number to a hexadecimal number:

1. Convert binary number to decimal number, and then convert to hexadecimal number.
2. Using grouping of bits.

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

For the integer part, make groups of four bits together starting from the left side of the radix point, and for the fractional part, make groups of four bits towards the right side of the radix point.

Replace each group with the corresponding hexadecimal number, we get the hexadecimal equivalent of the given binary number.

**NOTE:** Add any number of 0's in the left most bit for integer part, and 0's in right most bit for fractional part required to complete the grouping of 4 bits.

**Ques 1 - Convert binary number 1000110101110 to hexadecimal number.**

Converting Binary Number into Decimal Number:

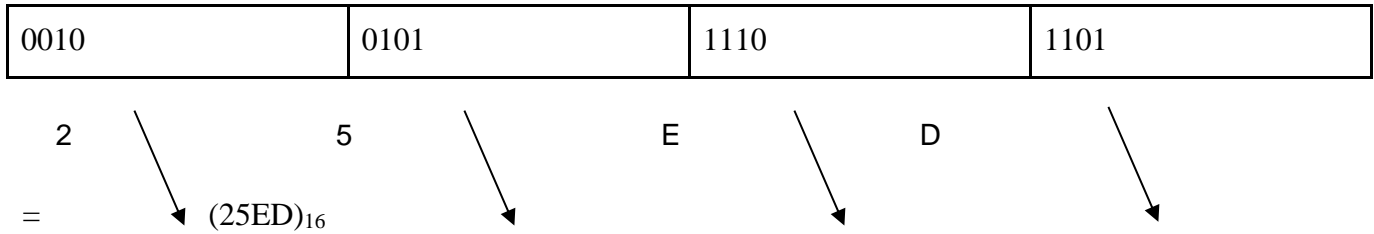
$$\begin{aligned}
 &= (1000110101110)_2 \\
 &= 1 \times 2^{12} + 0 \times 2^{11} + 0 \times 2^{10} + 0 \times 2^9 + 1 \times 2^8 + 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 \\
 &= 4096 + 0 + 0 + 0 + 256 + 128 + 0 + 32 + 0 + 8 + 4 + 2 + 0 \\
 &= (4526)_{10}
 \end{aligned}$$

Converting Decimal into Hexadecimal Number:

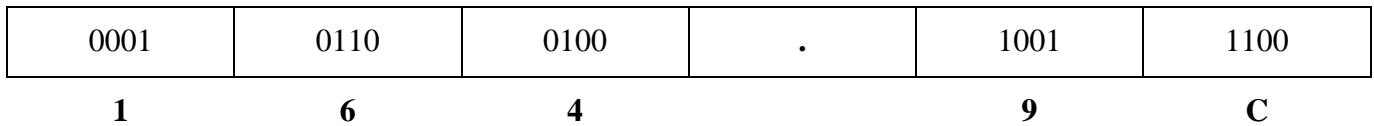
Divisor	Dividend	Remainder	
16	4526		
16	282	14(E)	
16	17	10(A)	
16	1	1	
	0	1	

$$= (11AE)_{16}$$

**Ques 2 - Convert Binary number 10010111101101 into Hexadecimal Number.**



**Ques 3 - Convert Binary number 101100100.100111 into Hexadecimal Number.**



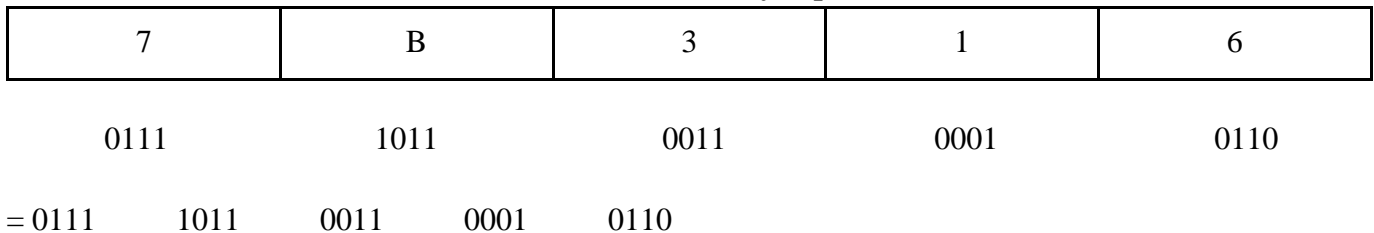
Hence the equivalent Hexadecimal Number is (164.9C)<sub>16</sub>

### Hexadecimal Number System to Binary Number System

A hexadecimal number can be converted to Binary Number through any of the two methods:

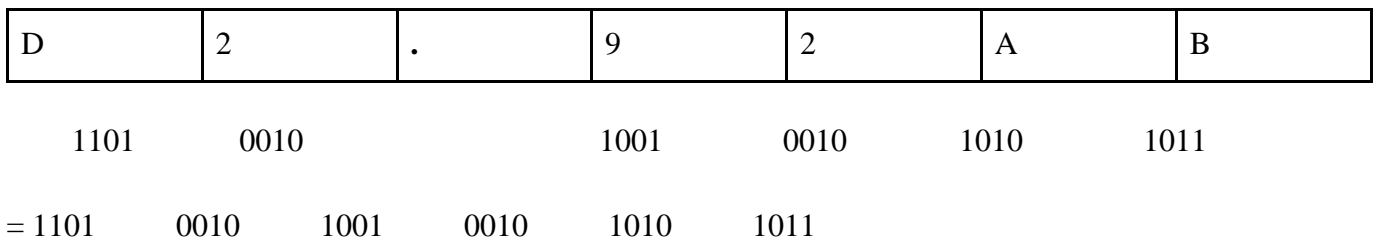
1. Converting Hexadecimal Number to Decimal Number, and then converting Decimal Number to Binary number.
2. Using Conversion Table

**Ques 1 - Convert Hexadecimal Number 7B316 to Binary Equivalent Number.**



Hence the equivalent binary number is (01111011001100010110)<sub>2</sub>

**Ques 2 - Convert Hexadecimal Number D2.92AB to Binary equivalent Number.**



Hence the equivalent binary number is (11010010.1001001010101011)<sub>2</sub>

## Binary Number System to Octal Number System

There are two ways to convert a binary number to a hexadecimal number:

1. Convert binary number to decimal number or octal number, and then convert to octal number.
2. Using grouping of bits

Binary Number	Octal Number
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

For the integer part, make groups of three bits together starting from the left side of the radix point, and for the fractional part, make groups of three bits towards the right side of the radix point.

**NOTE:** Add any number of 0's in the left most bit for integer part, and 0's in the right most bit for fractional part required to complete the grouping of 3 bits.

**Ques 1 - Convert binary number 1011011010 to octal number.**

001	011	011	010
1	3	3	2

= (1332)<sub>8</sub>

## Octal Number System to Binary Number System

An Octal number can be converted to Binary Number through any of the two methods:

1. Converting Octal Number to Decimal Number, and then converting Decimal Number to Binary number.
2. Using Conversion Table







## Encoding Schemes

Encoding is defined as the process to convert data from one form to another. Computers only understand binary language. There is a need to convert popularly used languages by humans into machine understandable format. Textual characters (letters, numbers and symbols) are assigned unique numerical codes. Some of the popular encoding schemes are ASCII, ISCII and Unicode.

### ASCII

- ASCII stands for American Standard Code for Information Interchange.
- It was developed in the United States of America by American Standards Association (ASA).
- It is used to represent textual information in computers.
- ASCII uses 7-bits for encoding.
- A maximum of 128 characters may be encoded. Out of these 128 characters, only 95 are printable including the digits 0-9, lowercase and uppercase characters a-z A-Z and punctuation marks. The rest of the characters are Control Characters.

### ISCII

- ISCII stands for Indian Script Code for Information Interchange (ISCII).
- ISCII is an extended version of the ASCII code and uses 8 bits for encoding.
- It was developed by the Bureau of Indian Standards, India.
- It represents various languages from India.
- Some of the supported scripts are Devanagari (Hindi, Marathi, Sanskrit, Konkani), Gujarati, Kannada, Punjabi, Malayalam, Telugu, Tamil, Oriya, Bengali-Assamese etc.

### Unicode

- Unicode Standard is developed and managed by the Unicode Consortium, which is a non-profit organization composed of members from Software Development companies such as Apple, Adobe, Google, IBM, Microsoft etc.
- It aims to provide a universal platform for encoding characters of various languages from the world.
- It assigns a unique Code Point to every character, independent of the CPU architecture/platform, or the underlying software.
- Unicode covers most writing scripts across the world. As of now, over 161 scripts are included in the latest version of Unicode.
- Unicode defines two mapping methods: the Unicode Transformation Format (UTF) encodings, and the Universal Coded Character Set (UCS) encodings.
- Various versions of UTF are UTF-8, UTF-16, UTF-32.
- All UTF encodings map code points to a unique sequence of bytes.

- UTF-8 uses 8-bits or 16-bits or 24-bits or 32-bits for each code point.
- UTF-16 uses 16-bits or 32-bits for each code point.
- UTF-32 uses 32-bits for each code point.

## Questions and Answers

### Very Short Answer Questions

**Q. What is a computer?**

A. Computer is an electronic device that takes instructions and data from a user, processes them, and produces some meaningful output as a result.

**Q. What is hardware?**

A. Hardware are physical components which can be seen and touched (tangible). Examples of hardware include CPU, mouse, keyboard, monitor, motherboard, cables, CPU case, power supply unit, RAM, graphic card, sound card etc.

**Q. What is software?**

A. Software is a set of instructions, its documentation and data, which are stored digitally on the computer. The software is intangible, i.e cannot be touched. Examples are Microsoft Windows, Linux, Paint, Word, PowerPoint, Photoshop, VLC Media Player, VS Code, Python.

**Q. What is an Operating System?**

A. An operating system is a system software that acts as an interface between the user and the computer hardware and manages all the resources of a computer.

**Q. Give 2 examples of Mobile OS and Desktop OS each.**

A. Mobile OS - Android & Apple iOS. Desktop OS - Microsoft Windows, Ubuntu.

**Q. What is a mouse?**

A. The mouse is an input device responsible for the movement of the cursor on the computer's screen.

**Q. What is a keyboard?**

A. The keyboard is an input device based on the traditional typewriter, which uses an arrangement of keys. Keyboards come in various layouts and languages.

**Q. What is a scanner?**

A. Scanner is an input device that optically scans and digitally captures images from physical photographic prints, documents, posters etc. Commonly used office scanners use light beams to scan documents placed on a glass flatbed.

**Q. What is a volatile memory?**

A. The memory where the contents get erased as soon as the power is turned off is called volatile memory.

**Q. What is a non-volatile memory?**

A. The memory where the contents do not get erased when the power is turned off is called non-volatile memory.

**Q. What is the full form of BIOS?**

A. Basic Input Output System.

**Q. What are the two types of primary memory?**

A. Two types of primary memory are RAM and ROM.

**Q. How many binary digits are there?**

A. There are two binary digits, 0 and 1.

**Q. What is a nibble?**

A. A sequence of 4-bits is called a nibble.

**Q. What is a byte?**

A. A sequence of 8 bits is called a Byte (B).

**Q. What does ISCII stand for?**

A. ISCII stands for Indian Script Code for Information Interchange.

**Q. What are the base values of the octal and hexadecimal number system?**

A. The Octal number system uses base-8 whereas the Hexadecimal number system uses base-16.

**Q. Write 2 examples of High Level Programming Languages.**

A. Python and C++.

**Q. Write 2 examples of Low Level Programming Languages.**

A. Binary Language and Assembly Language.

**Q. What are the different types of RAM?**

A. RAM can broadly be classified in two types, Static RAM (SRAM) and Dynamic RAM (DRAM).

**Q. What are the different types of ROM?**

A. ROM can be classified into PROM, EPROM and EEPROM.

**Q. Differentiate between Data and Information.**

A. Data is any sequence of one or more symbols. In other words, raw facts and figures constitute data. Whereas, Data when processed or interpreted is known as Information.

**Q. What is Encoding?**

A. Encoding is a system that maps characters, symbols, or textual elements to numeric values or bit patterns for digital representation and storage. It is used to represent human-readable text and symbols as binary data, which can be processed, transmitted, and stored in computers and other digital systems.

**Q. Which number system is used in our day to day life?**

A. Decimal Number System.

**Q. Differentiate between Program and Process.**

A. A Program is a set of instructions stored in the memory whereas a process is a program under execution.

**Q. Write 4 examples of Program Development Tools.**

A. Microsoft VS Code, Sublime Text, PyCharm, Spyder.

**Q. What are Registers?**

A. Registers are local or temporary storage for the CPU. They are present on the CPU itself and can be accessed quickly.

**Short Answer Questions**

**Q. What are the components of a computer?**

A. The major components of a computer are the Central Processing Unit (CPU), Memory (RAM, and ROM), Storage devices (HDD, SSD, Optical Drives, Flash Drives), Input Devices, Output Devices, Motherboard and the Power Supply Unit.

**Q. What is the difference between Hardware and Software?**

A. The devices that are tangible in nature (can be touched) are called Hardware Devices, whereas Software being digital cannot be touched. Hardware is a physical component of a computer, whereas Software is a set of instructions to a computer to perform a specific task.

Hardware includes Input and Output Devices, Storage Devices, Internal tangible Components of a Computer. Software includes Programming Software, Application Software, Operating system etc.

**Q. What are the components of the CPU?**

A. The CPU is made up of Arithmetic Logic Unit (ALU), Control Unit (CU), Set of Registers, Cache Memory and the Bus Interface.

**Q. How does ALU work?**



**Q. What is the purpose of the system bus?**

A. The system bus provides a path facilitating the flow of data between the central processing unit (CPU), memory, and various peripheral devices. The system bus is typically divided into three main types of buses, Address Bus, Control Bus and the Data Bus.

**Q. Differentiate between compiler and interpreter.**

A. Both Compiler and Interpreter are language translators that convert High Level Programming Language into Low Level Language. A compiler converts the source code into Low Level Language all at once, whereas an interpreter translates the source code line by line.

Example of Compiler: GCC Compiler

Example of Interpreter: Python Interpreter

**Q. What is the primary objective of an Operating System?**

A. The primary objective of an operating system is to provide services for building and running application programs, and to provide a user interface.

**Q. What is primary memory?**

A. Primary memory is the computer's immediate and directly accessible storage that holds data and instructions that the central processing unit (CPU) can quickly access during program execution. All instructions and data has to be loaded onto the primary memory before it can be sent to the CPU for processing.

**Q. What is Cache memory?**

A. A very high speed, small sized memory placed between CPU and primary memory is known as Cache Memory. It stores frequently accessed data, thus reducing access time and speeding up overall performance of the system.

**Q. What is secondary memory?**

A. Secondary Memory is a non volatile memory used to store data and programs. It is also known as Auxiliary Memory. It has larger storage capacity than primary memory, but slower and cheaper than the main memory. Examples of secondary memory are - Hard Disk Drive, Memory Card, CD/DVD, Magnetic Tapes etc.

**Q. Explain the types of system software.**

A. The System Software is further classified as - Operating Systems, System Utilities and Device Drivers.

**Operating Systems** - It manages and controls various resources of the computer system and provides an user-friendly environment to interact with the hardware of the computer.

**System Utilities** - These are the programs used for maintenance and configuration of the computer system. These are used to monitor system performance, manage files and storage, diagnosing issues, and maintaining system security.

**Device Drivers** - It is an interface between a particular device and the operating system. It ensures proper functioning of the device, overall management of a particular device.

**Q. What are the types of programming languages?**

A. There are two types of programming languages - High Level Language and Low Level Language  
Low Level Languages are the languages that are close to the hardware architecture. These are used for system level programming, device drivers, embedded systems.

High Level Languages are the languages that are close to natural human languages, and not close to the hardware.

Examples of Low Level Languages are Machine Language, Assembly Language

Examples of High Level Languages are C++, Java, Python etc.

**Q. What is the difference between Operating System and Device Driver?**

A. A device driver facilitates the interaction between the OS and the hardware devices. Whereas, the OS is responsible for the interaction between the user and the hardware.

**Q. What is the reason for grouping the 4 bits in a binary number to form a Hexadecimal number?**

A. The hexadecimal number system uses 16 unique symbols (base-16) for representing numbers, and with 4 bits, each digit in the hexadecimal system can be represented uniquely. Therefore, 4-bit binary sequences are sufficient to represent all possible symbols in the hexadecimal number system.

**Q. Which input device is most suitable for the OS with the following user-interface (UI)?**

1. Graphical UI
2. Command Line UI
3. Gesture Based Interface
4. Voice Based Interface
5. Touch Based Interface

- ANS. 1. Mouse  
2. Keyboard  
3. Camera/Webcam  
4. Microphone  
5. Touch-screen

**Q. Why is an Operating system called a Resource Manager?**

A. The OS is responsible for managing various hardware and software resources in a computer system, therefore it can be called a resource manager. It performs the following functions, Memory management, CPU Scheduling, Device management, Storage management, Network Management etc.

**Q. What is a HDD and SSD?**



A. HDD stands for Hard Disk Drive. It is a mechanical non-volatile secondary storage device. SSD stands for Solid State Drive, which is a non-volatile memory that stores data without moving parts. SSDs are faster, more durable, more compact, quieter, and consume less energy.

**Q. What is OCR?**

A. OCR stands for Optical Character Recognition. It is a pattern recognition technology that identifies the text present in digital form, such as an image, document etc.

**Q. What is MICR?**

A. MICR stands for Magnetic Ink Character Reader. It is a device that reads a special kind of ink which is sensitive to magnetic fields. It is often used to verify the originality of cheques and other documents.

**Q. What are some of the applications of the Hexadecimal Number System?**

A. Hexadecimal numbers are commonly used in various applications due to their compact representation and ease of conversion to and from binary. They are also used to represent addresses in memory, to represent colors on web pages, to display error messages etc.

**Q. Write a few advantages of computers.**

A. Few of the advantages of computers are as follows,

- Multi-tasking
- Speed
- Productivity
- Accuracy
- Connectivity
- Education
- Storage
- Reliability

### **Long Answer Questions**

Q. Explain the organization of computer systems.

Q. What are input devices? List 5 input devices.

Q. What are output devices? List 5 output devices.

Q. What is the IPO Cycle?

Q. Explain the functions of the Operating System.

- Q.** Explain different types of OS User Interface.
- Q.** Explain the difference between primary memory and secondary memory.
- Q.** What is the need of Cache memory?
- Q.** How are high level languages converted into low level languages?
- Q.** Write down some advantages of computers.
- Q.** Convert the following numbers from one number system to another:

- |                             |                                       |
|-----------------------------|---------------------------------------|
| 1. $(945)_{10} = (?)_2$     | <b>Ans.</b> $(1110110001)_2$          |
| 2. $(42)_{10} = (?)_8$      | <b>Ans.</b> $(52)_8$                  |
| 3. $(194)_{10} = (?)_{16}$  | <b>Ans.</b> $(C2)_{16}$               |
| 4. $(10110)_2 = (?)_{10}$   | <b>Ans.</b> $(22)_{10}$               |
| 5. $(745)_8 = (?)_{10}$     | <b>Ans.</b> $(485)_{10}$              |
| 6. $(E1A)_{16} = (?)_{10}$  | <b>Ans.</b> $(3610)_{10}$             |
| 7. $(0010111)_2 = (?)_{16}$ | <b>Ans.</b> $(17)_{16}$               |
| 8. $(6FA2)_{16} = (?)_2$    | <b>Ans.</b> $(0110111110100010)_{16}$ |
| 9. $(101111)_2 = (?)_8$     | <b>Ans.</b> $(57)_8$                  |
| 10. $(532)_8 = (?)_2$       | <b>Ans.</b> $(101011010)_2$           |
| 11. $(C45D)_{16} = (?)_8$   | <b>Ans.</b> $(142135)_8$              |
| 12. $(165)_8 = (?)_{16}$    | <b>Ans.</b> $(75)_{16}$               |

### Fill in the Blanks

- The binary language has two digits namely 0 and 1.
- The full form of the CPU is the Central Processing Unit.
- The full form of ALU is the Arithmetic Logic Unit.
- The full form of CU is the Control Unit.
- Set of Registers are temporary and local storage for the CPU.
- The System Bus provides a way for the CPU to communicate with the connected devices.
- Encoding Schemes transforms text into number codes that facilitate the communication among computers.
- Unicode is the encoding standard that is capable of representing multiple scripts.
- The CPU is the brain of the computer responsible for executing instructions.
- RAM stands for Random Access Memory
- The CPU register is a temporary storage area that holds data and instructions while they are being processed.

12. The cache memory stores the most frequently used data and instructions to improve system performance.
13. The ROM (Read-Only Memory) is a non-volatile memory that retains its data even when the power is turned off.
14. The central processing unit (CPU) consists of the control unit and the arithmetic logic unit (ALU).
15. The system bus is responsible for managing communication between the CPU and other components.
16. The primary purpose of an operating system is to manage computer resources.
17. A group of eight bits is known as a byte.
18. An instruction is a basic arithmetic or logical operation that the CPU can perform.
19. The physical components of a computer system are referred to as hardware.
20. A cache is a small, high-speed memory located close to the CPU that stores frequently accessed data.
21. A software is a collection of programs and data that enables a computer to perform a specific task.

### State True or False

- Q. Microphone is an Input Device. (True)
- Q. Hardware cannot be seen or touched physically. (False)
- Q. Software cannot be seen or touched physically. (True)
- Q. Operating System is the interface between the User and the Hardware. (True)
- Q. MICR stands for Magnetic Ink Character Reader. (True)
- Q. ALU stands for Arithmetic and Logic Unit. (True)
- Q. Registers are permanent storage. (False)
- Q. Computer memory is organized into memory cells, each of which stores a fixed amount of data, typically represented in binary as 0s and 1s. (True)
- Q. RAM is categorized as DRAM and SRAM. (True)
- Q. EEPROM stands for Electronic Erasable Primary ROM. (False)
- Q. A sequence of 4-bits is called a byte. (False)
- Q. The compiler converts high level language into low level language at once. (True)
- Q. Gesture Based User Interface allows users to perform tasks without physically touching the system. (True)

- Q. Boolean OR Operator results True if both the inputs are true, otherwise false. (False)
- Q. A logic circuit is an electronic circuit used in computers to perform a logical operation using Logic Gates. (True)
- Q. The Control Unit is responsible for Arithmetic and Logical Operations. (False)
- Q. System Bus is a pathway for signals to travel from one location to another in the Computer System. (True)
- Q. The CPU directly interacts with the Secondary Memory. (False)
- Q. Typically, RAM is faster than Hard Disk Drives. (True)
- Q. RAM is larger in size than the Secondary Memory. (False)
- Q. Static RAM needs constant refreshing to preserve the data. (False)
- Q. RAM is non-volatile in nature. (False)
- Q. ROM stores the software that is used to start-up the computer. (True)
- Q. BIOS stands for Basic Input Output System. (True)
- Q. Cache memory reduces the access time of RAM by storing the frequently accessed items. (True)
- Q. The base of binary, octal, decimal and hexadecimal number systems are 2,8,10 and 16 respectively. (True)
- Q. A binary number is made of a sequence of bits. (True)
- Q. 1 KiloByte (KB) = 1024 MegaBytes (MB) (False).
- Q. Each hexadecimal digit has a unique 4-bit representation. (True)
- Q. 1 MegaByte (MB) = 1024 KiloBytes (KB) (True)
- Q. Interpreter is a System Utility software. (False)
- Q. Compiler is a Language Translator Software. (True)
- Q. Microsoft Windows is an example of Application Software. (False)
- Q. Device Drivers translate code from high level language to low level language. (False)
- Q. Compiler and Interpreter convert high level languages to low level languages. (True)
- Q. Command Line User Interface uses Graphical Icons and Menus for navigation. (False)

**Multiple Choice Questions**  
**(Choose any one out of four choices)**

1. **Which of the following translators converts high level language into low level language at once?**
  - a. Assembler
  - b. Compiler**
  - c. Interpreter
  - d. None of the above
  
2. **Which of the following is a System Utility?**
  - a. Windows
  - b. iOS
  - c. Symbian
  - d. Anti-Virus**
  
3. **Which of the following is not a role of the Operating System?**
  - a. Memory Management
  - b. Storage Management
  - c. Language Translation**
  - d. Process Management
  
4. **Which of the following is an Output Device?**
  - a. Mouse
  - b. Keyboard
  - c. Speakers**
  - d. Joystick
  
5. **Which of the following is not an Operating System?**
  - a. Microsoft Windows
  - b. Instagram**
  - c. Ubuntu
  - d. Apple macOS
  
6. **Which of the following is a Language Translator?**
  - a. Compiler**
  - b. Defragmentation Tool
  - c. Anti-Virus
  - d. System Restore Utility
  
7. **1 MB is equal to**
  - a. 1024 GB
  - b. 1024 MB

- c. **1024 KB**
  - d. 1024 B
8. A symbol in the binary number system can assume how many values?
- a. **2**
  - b. 7
  - c. 8
  - d. 10
9. Nibble is a sequence of bits of length
- a. 1
  - b. 2
  - c. 3
  - d. **4**
10. Which of the following is not an example of High-Level Programming Language?
- a. Java
  - b. Python
  - c. **Binary Language**
  - d. C++
11. \_\_\_\_\_ is a non-volatile memory
- a. RAM
  - b. **ROM**
  - c. EPROM
  - d. EEPROM
12. \_\_\_\_\_ memory requires constant refreshing to preserve its contents.
- a. ROM
  - b. PROM
  - c. **DRAM**
  - d. SRAM
13. Which of the following is not a type of Memory?
- a. Primary
  - b. Secondary
  - c. **Tertiary**
  - d. Cache
14. IPO stands for:
- a. **Input-Process-Output**
  - b. Input-Program-Output

- c. Instruction-Program-Outcome
- d. Information-Process-Outcome

15. Which of the following is not a component of the CPU?

- a. ALU
- b. DVD**
- c. CU
- d. Registers

16. Which of the following is not an advantage of Computers?

- a. Multi-tasking
- b. Reliability
- c. Accuracy
- d. High Cost**

17. The number  $(1000)_2$  in decimal number system is equivalent to

- a. 8**
- b. 4
- c. 16
- d. 7

18. Arrange in ascending order of size:

- a. MB, KB, Byte, Nibble
- b. Nibble, KB, MB, GB**
- c. Bit, MB, Nibble, KB
- d. Bit, GB, KB, MB

19. Whenever the computer is started or a software application is launched, the required program is loaded into \_\_\_\_\_ for processing.

- a. ROM
- b. CPU
- c. RAM**
- d. CU

20. If S is a Boolean variable, determine the incorrect Boolean statement

- a.  $S + 1 = S$**
- b.  $S + 1 = 1$
- c.  $S + 0 = S$
- d.  $S + S = S$

21. ASCII stands for \_\_\_\_\_

- a. American Standard Coding For Information Interdiscipline
- b. American Standard Code For Information Initiative
- c. American Standard Code For Information Interchange**
- d. American Simple Code For Information Interchange

22. **The output of the two-input NAND gate is 1**

- a. if one input is 1 and the other is 0
- b. if both the inputs are 1
- c. if at least one input is 0
- d. if both the inputs are 0**

23. **XOR is represented as**

- a.  $A.B$
- b.  $A+B$
- c.  $AB' + A'B$**
- d.  $AB + A'B'$

### **ASSERTION (A) AND REASON (R) BASED QUESTIONS**

**Mark the correct choice as**

- 1. Both A and R are true and R is the correct explanation for A.
- 2. Both A and R are true but R is not the correct explanation for A.
- 3. A is True but R is False.
- 4. A is False but R is True.

Q.

Assertion (A): Computers perform complex tasks with incredible speed and accuracy.

Reason (R): The smallest unit of memory is called a bit.

**Answer: Both A and R are true but R is not the correct explanation for A.**

Q.

Assertion (A): File Management is one of the roles performed by the OS of a system.

Reason (R): An Operating System is an interface between the user and the hardware and manages all the resources of a computer.

**Answer: Both A and R are true and R is the correct explanation for A.**

Q.

Assertion (A): NAND and NOR are called the Universal Gates.



Reason (R): OR operator results True(1) if both the inputs are True(1), otherwise False(0).

**Answer: A is True but R is False.**

Q.

Assertion (A): Each symbol in a hexadecimal number system can be represented by a unique 4-bit binary number.

Reason (R): The hexadecimal number system has 16 unique symbols.

**Answer: Both A and R are true and R is the correct explanation for A.**

Q.

Assertion (A): Computers understand our language, and hence there is no need to convert into machine language.

Reason (R): Encoding is defined as the process to convert data from one form to another.

**Answer: A is False but R is True.**

Q.

Assertion (A): The software that allows maintenance and configuration of the system is called System Utility.

Reason (R): MS windows is an example of System Utility.

**Answer: A is True but R is False.**

## Unit II

# Computational Thinking and Programming - I

### Problem and Problem Solving

In computer science, "problem" refers to a task or challenge that requires a solution. The process of identifying a problem, developing an algorithm, and implementing an algorithm to develop a computer program is called Problem Solving. Computers may be used to solve various daily life problems such as Train Ticket Booking, Online Shopping and Net-Banking etc.

### Steps required for solving a problem

- Analyzing the problem
- Developing an Algorithm
- Coding
- Testing and Debugging

### Analyzing the Problem

This stage focuses on understanding the problem. If we do not have a clear understanding of the problem, we may develop a computer program that cannot solve the problem correctly. In this stage, we figure out the inputs, the outputs and the processing required to convert the input into the output.

### Developing Algorithm

This stage focuses on creating a logical sequence of instructions, called an Algorithm. The algorithm can be executed by a computer to generate the desired output. An algorithm has a distinct start and end point, as well as a defined number of steps. For a given problem, more than one algorithm may be possible and the most suitable algorithm may be chosen.

### Algorithm for finding whether a number is Even or Odd

START

Step 1 → Take an integer number A

Step 2 → Divide A by 2, and store the remainder as r

Step 3 → If r is equal to 0, A is an Even Number

Step 4 → Else it is an Odd Number

STOP

### Algorithm for finding whether a number is a Prime number or Not

START

Step 1 → Take an integer number A

Step 2 → Set divisor as 2

Step 3 → Set flag\_value as True

Step 4 → Repeat from divisor to A-1  
    4a. divide A by divisor and store the remainder as r  
    4b. If r is zero, set flag\_value to False  
    4c. Increment divisor by 1  
Step 3 → If flag\_value is False, A is not prime  
Step 4 → Else A is prime  
STOP

## Coding

Coding is the process of creating computer programs.

## Testing

Testing is a process to check if an application is working as expected (and not working abnormally). The main objective of Testing is to find errors.

## Debugging

Debugging is the activity to fix the errors found in the application during the testing phase.




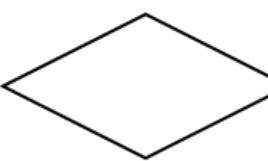

## Representation of Algorithms

There are two common methods of representing an algorithm —flowchart and pseudocode.

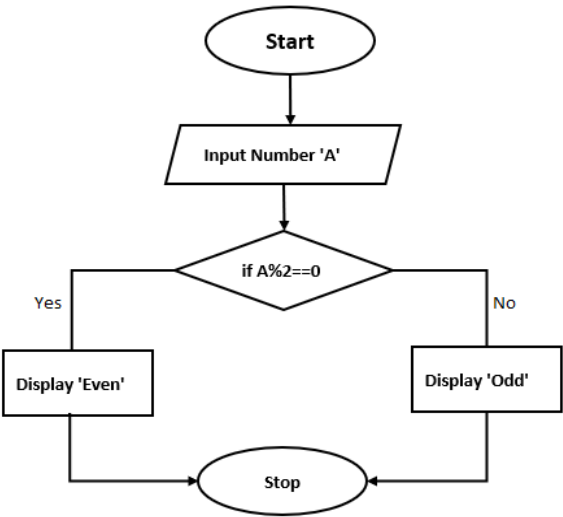
## Flowchart

- A flowchart is a graphical representation of an algorithm.
- A flowchart contains various shapes which are connected by arrows, which shows the flow of control.

## Shapes used in Flowchart

Shape	Shape Name	Usage
	Oval	Start/Stop
	Rectangle	Process
	Parallelogram	Input/Output
	Diamond	Decision/Condition
	Arrow	Flow of Control / Connections

**Draw a flow-chart to identify whether a number taken as the input from the user is an even number or an odd number?**

Algorithm	Flow Chart
<p>START</p> <p>Step 1 → Take an integer number A as input</p> <p>Step 2 → Divide A by 2, and store the remainder as r</p> <p>Step 3 → If r is zero, Display 'Even'</p> <p>Step 4 → Else Display 'Odd'</p> <p>STOP</p>	 <pre> graph TD     Start([Start]) --&gt; Input[/Input Number 'A'/]     Input --&gt; Decision{if A%2==0}     Decision -- Yes --&gt; DisplayEven[Display 'Even']     Decision -- No --&gt; DisplayOdd[Display 'Odd']     DisplayEven --&gt; Stop([Stop])     DisplayOdd --&gt; Stop     </pre>

## Pseudocode

- Pseudocode is a way of representing an algorithm in readable and easy language.
- Pseudocode is not an actual program. So, it cannot be executed.
- Some of the frequently used keywords while writing pseudocode are INPUT, COMPUTE, PRINT IF/ELSE, START, STOP

## Advantages of Pseudo-Code:

1. Easily convertible to a Programming Language
2. Easy to understand and read

## Write a pseudocode for identifying if a number is even or odd?

```
INPUT number A
COMPUTE remainder as r = A%2
IF r ==0 PRINT 'Even'
ELSE PRINT 'Odd'
```

## Decomposition

Decomposition is the process of breaking a complex computer problem into smaller parts that are easily manageable and solvable.

---

## Familiarization with the basics of Python programming

### Computer Program:

A computer program is a set of instructions that can be executed by the computer to perform and solve a certain task. These programs are written in a special language known as Programming Language.

### Computer Programming:

It is the process to create a computer program.

### Python Programming Language:

Python is an interpreted, high-level programming language. It was developed by Guido van Rossum. It is user-friendly and is most popular for its easy-to-use syntax and readable code.

### Features of Python

- High-level Programming language.
- Interpreted language (as Python programs are executed by an interpreter)

- Easy to use
- Simple Syntax
- Python programs are generally written with fewer Lines of Code as compared to other programming languages.
- Case-sensitive. For example, 'NUMBER' and 'number' are treated differently in Python.
- Portable programming language - has ability run programs on many computer architectures and operating systems.

*Syntax: Set of rules for framing a valid statement in a programming language.*

## Working in Python

- Install Python on the computer (<https://www.python.org/downloads/>). Refer Appendix for installation instructions.
- Use Python IDLE (Integrated Development and Learning Environment) for developing python Programs.

## How to Display Data

print( ) function is used to print a message on the screen.

## A Simple Hello World Program

```
print("Hello World")
>>> print("Hello World")
Hello World
```

## Modes of working in Python

- Interactive Mode
- Script Mode

## Interactive Mode

In Interactive Mode, a python statement is executed in a command-line shell. It provides instant feedback for each statement while keeping prior statements in memory.

## Script Mode

In script mode, the instructions are saved in a file with a '.py' extension and executed from the beginning of the file. This mode is suitable for creating and running large programs. In script mode, all commands are stored in the form of a program or a script.

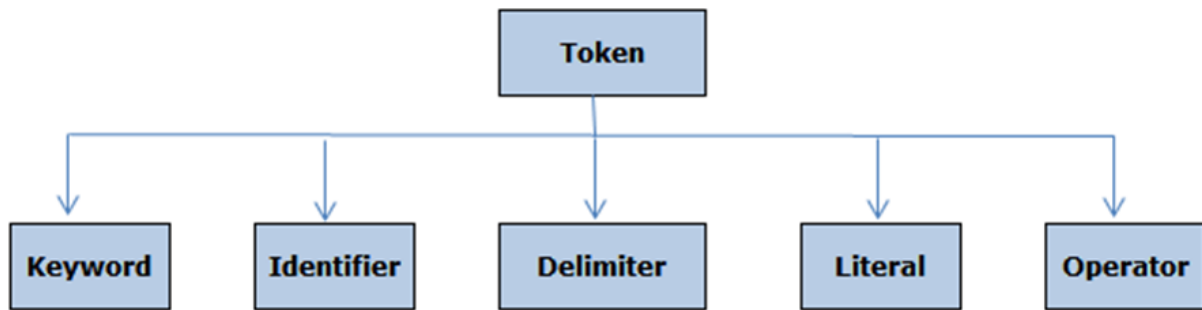
## Character Set:

A character set is a collection of valid characters that can be recognized by a language. Python Language recognises the following characters as valid:

Letters	: A-Z, a-z
Digits	: 0-9
Special Symbol	: + / @ ! - = <> == etc.
Whitespaces	: '\n', '\t' etc.

## Tokens

Tokens are the smallest individual units of a program.



## Keywords

Keywords are reserved words with special meaning (known to the interpreter). These can not be used as identifiers in a program.

Example: for, while, else, if

## Identifier

A variable, function, class, module, or other objects are identified by a name known as identifier.

### Rules for naming an identifier:

1. First character of a variable can be an alphabet(A-Z or a-z) or an underscore(\_).
2. Next characters of a variable can be an alphabet(A-Z or a-z), an underscore(\_) or a digit.
3. Keywords cannot be used as variables.
4. First character cannot be a digit.
5. Special characters including white spaces are not allowed.

## Literals:

Literals are data-items that have a fixed value of a certain data type, such as a number, string, boolean, or None. They are also known as Constants.

Example :

String literals (Text)	: 'Hello World'
Numeric literals (Numbers)	: 3.14
Boolean literals (Truth Value)	: True/False
None Literal	: The None keyword is used to define a null value or absence of a value. None is different from 0.

## Operators and Operands

Operators are symbols (or keywords) that perform various operations on the operands. An operand is a variable or a literal on which the operation is performed.

Example:  $50 + 20$

Here 50 and 20 are operands, and + is the operator.

**Arithmetic Operators** : Arithmetic operators perform mathematical operations like addition, subtraction, multiplication, division, floor division, exponent and modulus. (+, -, \*, /, //, \*\*, %)

## Relational Operators:

Relational operators perform comparison between values. An expression having relational operators evaluates to either True or False. Example >, <, >=, <=, ==, !=

## Python Comments

Comments are descriptions about the code. They help other programmers understand the functionality of the code. Comments are ignored by the Python interpreter, and are only relevant to the programmer.

## Variable

Variables refer to an object (data items like int, float, list, dictionary etc) stored in the memory. Value stored in a variable can be changed during the program execution.

Rules for naming a variable are the same as the rules for naming an Identifier.

Example: `pri = 5`

Here pri is a variable with value 5.

**Valid variable name examples** : name, Age, \_total

**Invalid variable name examples** : print value (whitespace not allowed), 1\_gender (cannot start with a digit)

## Concept of L Value and R Value

In Python, the l-value refers to the left-hand side of an assignment operator, while the r-value refers to the right-hand side of an assignment operator.

The l-value is associated with a valid memory location in the computer. The memory location can be checked by using the `id()` function.

The r-value may be any valid expression which is executed by the interpreter.

Consider the following assignment statement:



$x = 5+2$

In the statement above, the l-value is 'x' as it is on the left-hand side of the = operator.

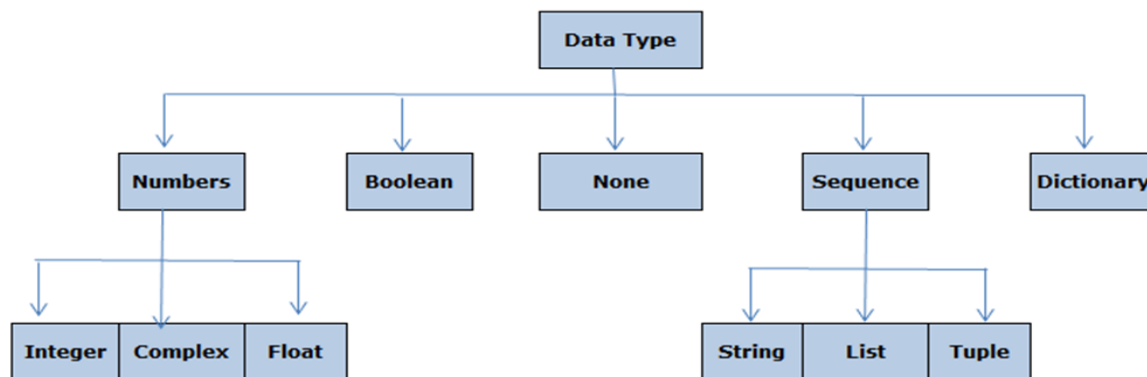
The r-value is 7 as it is on the right-hand side of the = operator, and is generated by performing the addition operation on 5 and 2.

The memory location of x may be checked by executing the command `id(x)`.

## Knowledge of Data Types

### Data Type

Data type represents the type of data a Variable or a Literal is referring to. Each data type has specific characteristics and operations associated with it. In Python, there are various data types, including number, string, boolean, list, tuple, and dictionary.



### Mutable and Immutable Data Objects

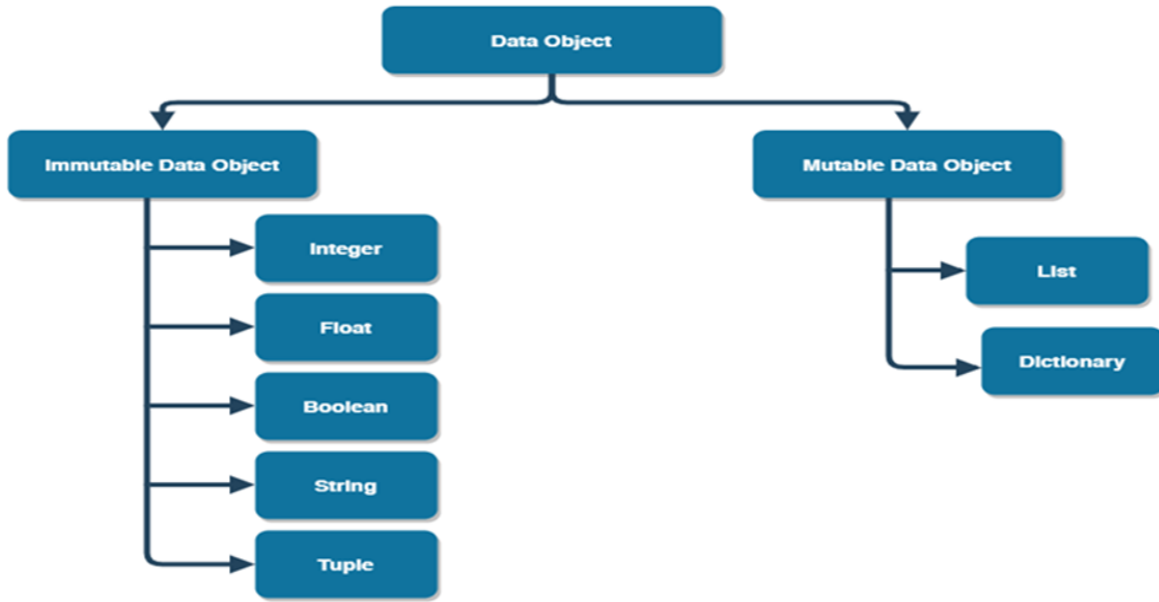
Python variables are memory references. It may be required to change or update the value of an object referenced by a variable. However, for certain data objects, Python does not allow us to change the value of an object.

#### Mutable Objects:

Mutable data objects are objects that can be changed after they are created. It is possible to add, remove, or modify elements within these data types. Example of mutable data types: List, Set and Dictionary.

#### Immutable Objects :

Objects whose values cannot be changed after they are created are called immutable objects. To change the value, a new object is created. Example of immutable data types: Number (Integer, Float), String, and Tuple.



## Numeric Data Types (Number)

Python has three numeric data types:

1. Integer
2. Float
3. Complex

### Integer

- Numbers with No Fractional Part
- Can be Positive or Negative
- In Python 3, the int type has no max limit. Values can be as large as the available memory allows.
- Example 100, 0o55 (octal number), 0x68(hexadecimal number)

### Float

- Numbers with Fractional Part
- Example 3.14, .314E01

### Complex

- Numbers with both real and imaginary components
- A complex number is represented by "x + yj". Example 10 + 9j

### Boolean

A boolean data type can assume one of the two possible values : True or False.

**Sequence:** Sequence is an ordered collection of items or elements which includes several built-in types as String, List, and Tuple. Values in the sequence are called elements/items. Each element in a sequence has a unique index.

### String:

- A string is an ordered sequence of characters enclosed in single/double/triple quotes.
- Single Line String : Terminates in a single line. Example – **'This is an example of single line'**
- Multi Line String : Does not terminate in a single line. A multiline string may be created using three quotes

Example -

```
'''This is a
   Multiline
   string'''
```

### List:

- List is an ordered sequence data type which can store values of any data type.
- List is mutable.
- List is enclosed in square brackets [ ]
- Example : [ ] is an empty List,  
**[5, 6.5, True, 'Hello']** is a List having 5, 6.5, True and 'Hello' as four elements.

### Tuple:

- Tuple is an ordered sequence which can store values of any data type.
- They are immutable, i.e the items of a Tuple cannot be updated after creation.
- Tuple is enclosed in parenthesis ( )
- Example : t=() is an empty Tuple  
t=(9,) is a Tuple with 9 as an item  
t=(**8, 5, 9.5, False**) is a Tuple with 8, 5, 9.5, False as four items

### Dictionary:

- Dictionary in Python stores items in the form of key-value pairs
- Syntax is dict\_variable = {key1:value1, key2:value2, ..., key-n:value-n}
- Items in a dictionary are enclosed in curly brackets { } and are separated by commas
- In a key-value pair, the key is separated from the value using a colon (:)
- To access any value in the dictionary, specify the key as the index using square brackets [ ].
- Example : { } is an empty dictionary,  
D = {'name':'Python','version':'3.7.2','OS': 'Windows'}  
**print(D['version'])** # shows 3.7.2 as output

## Special Data-type: None

The None data type/keyword is used to indicate absence of a value (No value or Missing Value).

**Operand:** Value(s) required for operation of operator.

**Operators:** Operators can manipulate the value of operands (variables or values). Various symbols are used as operators.

```
a = 5
b = 6
sum = a + b
```

In above example, + and = are the operators, a, b are operands and sum is a variable.

## Expression

An expression is combination of operators, operands, literals and parenthesis. An expression produces a value when evaluated.

## Python Statement

In Python, a statement is an instruction that the interpreter can execute.

## Types of Operators

1. Arithmetic Operators
2. Relational Operators
3. Logical Operators
4. Assignment Operators
5. Identity Operators
6. Membership Operators

**Arithmetic operators:** Arithmetic operators perform mathematical operations such as addition, subtraction, multiplication, division, and modulus.

Operator	Description	Example
+ Addition	Adds values on either side of the operator.	$a + b = 30$
- Subtraction	Subtracts right hand operand from left hand operand.	$a - b = -10$
* Multiplication	Multiplies values on either side of the operator	$a * b = 200$
/ Division	Divides left hand operand by right hand operand	$b / a = 2$
% Modulus	Divides left hand operand by right hand operand and returns remainder	$b \% a = 0$
** Exponent	Performs exponential (power) calculation on operators	$a ** b = 10$ to the power 20
//	Floor Division - The division of operands where the result is the quotient in which the digits after the decimal point are removed.	$9 // 2 = 4$ and $9.0 // 2.0 = 4.0$

## Modulo Operator

The modulo operator (%) evaluates the remainder of the operation when the operand on the left is divided by the operand on the right of the modulo operator.

Example

$$15 \% 7 = 1$$

$$13 \% 7 = 6$$

**Relational Operators:** Relational operators compare the operands. They evaluate to either True or False.

Relational Operator		
Symbol	Name	Syntax
>	Greater Than	$x > y$
<	Less Than	$x < y$
==	Equal To	$x == y$
!=	Not Equal To	$x != y$
>=	Greater Than or Equal To	$x >= y$
<=	Less Than or Equal To	$x <= y$

## Logical operators:

Logical operators combine logical values of true or false into a logical expression. They evaluate to either True or False. There are three basic types of logical operators: 'NOT', 'AND', and 'OR'.

Logical Operator		
Symbol	Name	Syntax
and	Logical AND	$x \text{ and } y$
or	Logical OR	$x \text{ or } y$
not	Logical NOT	$\text{not } x$

## Assignment operators:

Variables are assigned values using assignment operators.

Assignment Operator		
Symbol	Name	Syntax
=	Assignment	$x = y + z$
+=	Addition assignment	$x += y \text{ (} x = x + y \text{)}$
-=	Subtraction assignment	$x -= y \text{ (} x = x - y \text{)}$
*=	Multiplication assignment	$x *= y \text{ (} x = x * y \text{)}$
/=	Division assignment	$x /= y \text{ (} x = x / y \text{)}$

Example :

`sum = 5 + x`

`result += 5`

Here, the sum variable is assigned the sum of 5 and x. And, the result variable is assigned the sum of result and 5.

## Identity Operators

The identity operators in Python are "IS" and "IS NOT". They check whether the two objects are of the same data type and share the same memory address.

Example :

```
x = 5
y = 5
if x is y:
    print("x and y are the same object")
else:
    print("x and y are different object")
```

Output will display **x and y are the same object** because both of them reference the same memory location.

## Membership Operators

The membership operators in Python are 'IN' and 'NOT IN'. They check whether the operand on the left side of the operator is a member of a sequence (such as a list or a string) on the right side of the operator.

Example :

```
x = ['Red', 2, 'Green']
if 'Red' in x:
    print("Found")
else:
    print("Not Found")
```

Output will display **Found** because the item 'Red' is a member of the List x.

## Operator Precedence

Operator precedence determines the priority of operators in an expression. The precedence of various operators is shown in the Precedence Table.

For example,  $x = 7 + 3 * 2$ ; will result in 13, not 20, because operator  $*$  has higher precedence than  $+$ . The expression is evaluated as  $7 + 6$ .

## Associativity of Python Operators

Associativity refers to the order in which operators of the same precedence are evaluated. The associativity of an operator may be **left-to-right** or **right-to-left**.

### Precedence Table

Operator	Description	Associativity
()	Parentheses	Left-to-Right
**	Exponentiation	Right-to-Left
*, /, %, //	Multiply, divide, modulo and floor division	Left-to-Right
+, -	Addition and subtraction	Left-to-Right
<, <=, >, >=	Relational	Left-to-Right
in, not in is, not is	Membership, Identity	Left-to-Right
not	Logical AND	Right-to-Left
and	Logical OR	Left-to-Right
or	Logical NOT	Left-to-Right
=	Assignment	Right-to-Left

## Evaluation of Expression

### 1. Evaluate the expression $50 + 20 * 30$

Evaluation:

=  $50 + (20 * 30)$  #precedence of  $*$  is more than that of  $+$   
 =  $50 + 600$   
 = 650

### 2. Evaluate the expression $100 - 20 + 50$

Evaluation:

The two operators ( $-$ ) and ( $+$ ) have equal precedence and the associativity is from left to right so the left operator (i.e.  $-$ ) will be evaluated first.

=  $(100 - 20) + 50$   
 =  $80 + 50$   
 = 130

### 3. Evaluate the expression $9 + 3 ** 2 * 4 // 3$

Evaluation:

=  $9 + (3 ** 3) * 4 // 3$

```
= 9 + 27 * 4 // 3 (* and // has left to right associativity)
= 9 + 108 // 3
= 9 + 36
= 45
```

## Type conversion

It is the process of converting the value from one data type into another. Python supports two ways of Type conversion:

- Implicit conversion
- Explicit conversion

## Implicit conversion

This type of conversion is performed by Python Interpreter automatically without the user's intervention.

**Example:**

```
num1 = 20 # num1 is integer
num2 = 30.5 # num2 is float
sum1 = num1 + num2 # sum1 will use float to avoid
# loss of fractional part during addition

print(sum1)
print(type(sum1))
```

**Output:**

```
50.5
<class 'float'>
```

## Explicit Conversion:

This type of conversion is performed by the user manually. It is also known as type-casting. Explicit type-casting is performed using functions such as `int()`, `float()`, `str()` etc.

Syntax : `new_data_type (expression)`

**Example**

```
num1 = input("Enter a number : ") # takes a string input by default
var1 = int(num1) #converts string to integer
var1 = var1 * 3
print(var)
```

**Output**

```
Enter a number : 2
6
```



---

## Input and Output in Python

### Data Input:

`input()` function is used for getting input from the user. It returns the input as a String data type by default. Explicit type casting is required to convert the input string into any other data type if required.

### Data Output:

`print()` function is used for displaying output on the screen.

#### Example-1

```
num1 = int(input("Enter a Number : ")) # type casting the input from
String to Int
print("The Number is : ", num1)
```

#### Example-2

```
num = int(input("Enter a Number : "))
num_cube = num*num*num
print("The cube is : ",num_cube)
```

---

## Errors

An error is a problem that occurs in a program. Error sometimes halt the program execution, or produce unexpected results, and cause the program to behave abnormally.

- Syntax error
- Logical error
- Runtime error

### Syntax Error

Syntax are the rules for framing statements in a programming language. Any violations in the rules while writing a program are known as Syntax Errors. They prevent the code from executing and are detected by the Python interpreter before the execution of the program begins.

Some of the Common Syntax Errors are

- Parenthesis Mismatch
- Misspelled keyword
- Incorrect Indentation

## Logical Error

Logical errors occur when the code runs without any errors, but the output is not as expected. Logical errors are caused by a problem in the logic of the code.

Example : `Average = mark_1 + mark_2 / 2` # incorrect calculation of average marks

Corrected Code : `Average = (mark_1 + mark_2 ) / 2`

## Runtime Error

A runtime error causes abnormal termination of the program during the execution. Runtime error occurs when the statement is correct syntactically, but the interpreter cannot execute it.

**Example: 'division by zero'**

```
num1 = 5.0
num2 = int(input("num2 = ")) #if the user inputs zero, a runtime error
will occur
print(num1/num2)
```

---

## Flow of Control

Flow of Control refers to the order in which statements are executed in a program.

## Sequential Flow

The default control flow in a program is sequential flow, in which statements are executed line-by-line one after the other in a sequence in which they are written.

**Example**

```
x = 6
y = 7
z = y - x
print(z)
```

## Conditional Flow

Conditional flow refers to execution of certain statements only if a specific condition is met. This is accomplished by the use of conditional statements such as `if`, `if-else`, and `if-elif-else`.

### Example

```
num = int(input("Enter a number : "))
if(num>5):
    print("Number is greater than 5")
else:
    print("Number is less than 5")
```

#### Output

```
Enter a number : 55
Number is greater than 5
```

## Iterative Flow

Iteration means 'repetition'.

Iterative flow repeats statements in a block of code. Repetition or looping can be performed a fixed number of times or until a certain condition is met. This is accomplished through the use of iterative statements: **for** and **while**.

### Example-1

```
name = input("Enter your name : ")
for x in range(5): # range function creates a sequence of integers from 0 to 4
    print("Hello", name)
```

### Example-2

```
name = input("Enter your name : ")
i=1
while i<=5:
    print("Hello", name)
    i +=1
```

## Conditional Statements

Conditional statements execute specific blocks of code based on certain conditions. In Python, conditional statements are implemented using the keywords `if`, `if-else`, and `if-elif-else`.

### Terminology

#### Indentation

Indentation refers to the spaces at the beginning of a line.

Example

```
if a<5:  
    print(a)  
    print('Inner Block')  
print('Outside block')
```

#### Block of code

A block of code is a set of statements that are grouped together and executed as a single unit. A block of code is identified by the indentation of the lines of code.

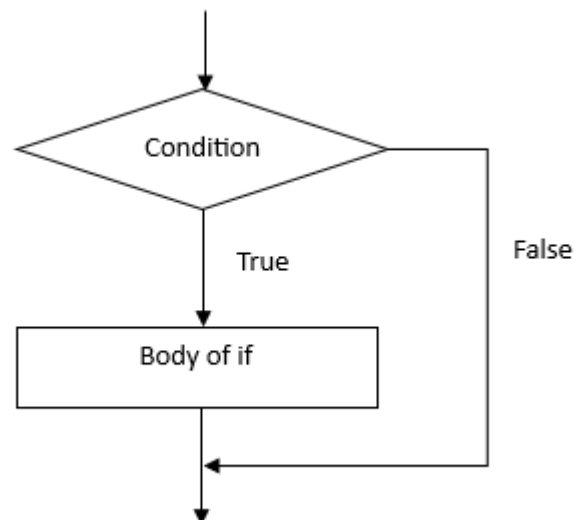
#### if statement

The `if` statement is used to execute a block of code only if a certain condition is true.

#### Syntax:

```
if <condition>:  
    Set of Statements
```

#### Flow Chart of if-statement



#### Example-

```
age = int(input("Enter your age : "))  
if age>=18:  
    print('Congratulations')  
    print('You are allowed to vote')
```

## Program

Write a python Program to find the absolute value of the number entered by the user.

```
num = int(input("Enter any number : "))
if num>=0:
    abs_num = num
else:
    abs_num= -num
print(abs_num)
```

### Output

Enter any number : -5

5

## Program

Write a python Program to sort three numbers entered by a user.

```
a = float(input("Enter the first number : "))
b = float(input("Enter the second number : "))
c = float(input("Enter the third number : "))
if a > b:
    a,b = b,a
if a > c:
    a,c = c,a
if b > c:
    b,c = c,b
print (a, "<", b, "<", c)
```

### Output

Enter the first number : 30

Enter the second number : 25

Enter the third number : 20

20.0 < 25.0 < 30.0

## if-else statement

The if-else statement is used to execute one block of code if a certain condition is true, and another block of code if the condition is false.

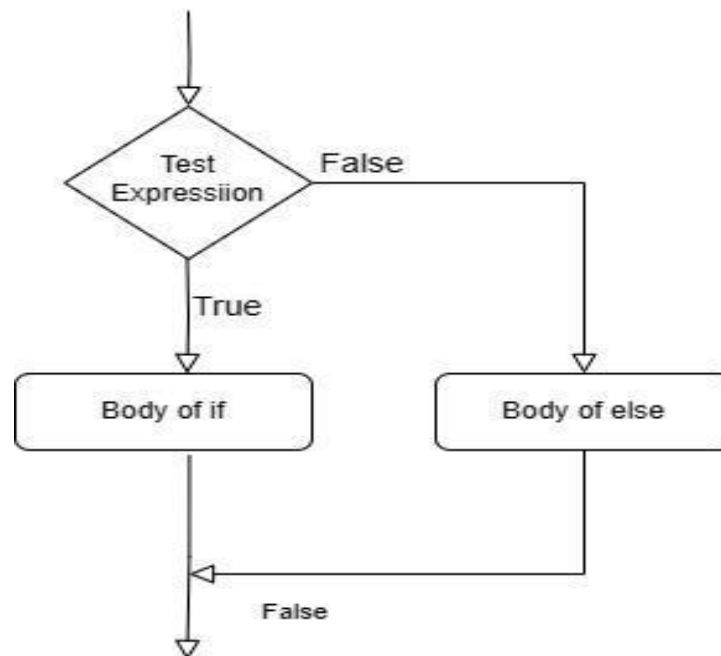
## Syntax of if-else

```
if <condition>:  
    set of statements  
else:  
    set of statements
```

## Example

```
age = int(input("Enter your age : "))  
if age >= 18:  
    print('Congratulations')  
    print('You are allowed to vote')  
else:  
    print('You are not allowed to vote')  
print('Thanks')
```

## Flowchart of if-else statement



## Program

Write a python Program to check if a number entered by a user is divisible by 3.

```
num=int(input("Enter a number : "))
if num % 3 == 0:
    print(num,"is divisible by 3")
else:
    print(num,"is not divisible by 3")
```

### Output

Enter a number : 601  
601 is not divisible by 3

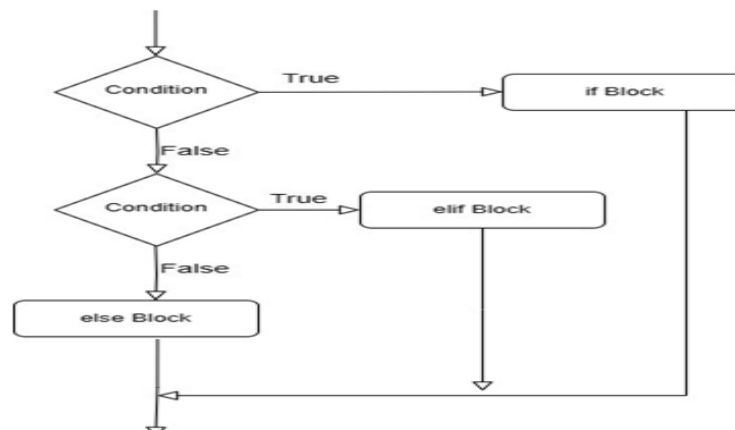
## if-elif-else statement

if-elif-else statement is used to check multiple conditions.

## Program

```
per = int(input("Enter Percentage : "))
if per >= 75:
    print("Distinction")
elif per >= 60:
    print("Grade-A")
elif per >= 50:
    print("Grade-B")
elif per >= 40:
    print("Grade-C")
else:
    print("Grade-D")
```

## Flowchart of if-elif-else statement



## Iterative Statement

Iterative statements execute a set of instructions multiple times. 'for' and 'while' loops are the iterative statements in Python.

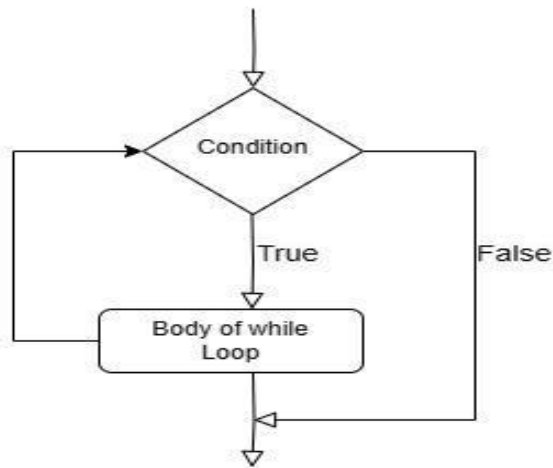
### while Loop

The while loop repeatedly executes a block of code as long as the specified condition is true. The while loop is an exit controlled loop, i.e. the user is responsible for exiting the loop by changing the value of the condition to False. While loop may run infinitely if the condition remains true.

#### Syntax:

```
while (condition):  
    block of statements
```

### Flowchart of while loop



### Example

Write a Python Program (using a while loop) to Find the Sum of First 10 Natural Numbers.

```
num=1  
sum=0  
while (num <= 10):  
    sum = sum + num  
    num = num+1  
print ("Sum of Natural Numbers : ", sum)
```

### Example

Write a Python Program (using the while loop) to Find the Sum of First N Natural Numbers where N is entered by the user.

```
num=1  
sum=0  
n = int(input("Enter the value of n : "))  
while (num <= n):  
    sum = sum + num  
    num = num+1  
print ("Sum of Natural Numbers : ", sum)
```



## range() Function

range() is a built-in function that returns a sequence of numbers.

### Syntax

`range(start, stop, step)`

start: The starting value of the sequence (inclusive). If not specified, it defaults to 0.

stop: The ending value of the sequence (exclusive).

step: The difference between two consecutive elements. Its default value is 1.

The range() function can be used in for loops to iterate over a sequence of numbers.

Case-1	Case-2	Case-3	Case-4
<code>x = list(range(5))</code> <code>print(x)</code>	<code>x = list(range(3,6))</code> <code>print(x)</code>	<code>x = list(range(3,20,2))</code> <code>print(x)</code>	<code>x = list(range(0, -9, -1))</code> <code>print(x)</code>
<b>Output</b> [0, 1, 2, 3, 4]	<b>Output</b> [3, 4, 5]	<b>Output</b> [3, 5, 7, 9, 11, 13, 15, 17, 19]	<b>Output</b> [0, -1, -2, -3, -4, -5, -6, -7, -8]

## for Loop

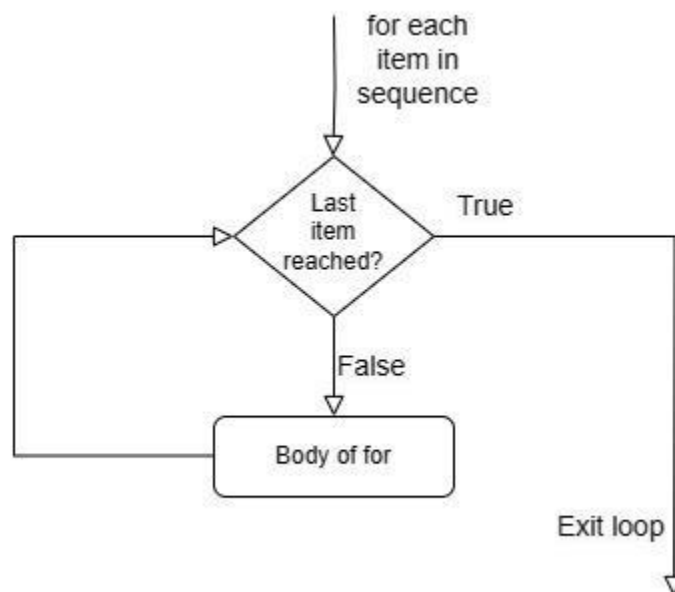
The for loop is used to iterate over a sequence (such as a list, tuple, string, etc.) and execute a block of code for each item in the sequence.

### Syntax

for <control\_variable> in <sequence>:

Block of code

## Flowchart of for loop



**Example-1**

```
list1 = [1,2,3,4,5,6,7,8,9,10]
for var1 in list1:
    print(var1)
```

**Example-2**

```
str1 = 'India'
for each_character in str1:
    print(each_character)
```

**Program-1**

Write a Python Program (using for loop) to Find the Sum of First 10 Natural Numbers.

```
sum=0
for num in range(1,11):
    sum = sum + num
print ("Sum of Natural Numbers : ", sum)
```

**Output:**

Sum of Natural Numbers : 55

**Program-2**

Write a Python Program (using for loop) to Find the Sum of First N Natural Numbers where N is entered by the user.

```
sum=0
n = int(input("Enter the value of n : "))
for num in range(1, n+1):
    sum = sum + num
print ("Sum of Natural Numbers : ", sum)
```

**Output:**

Enter the value of n : 20

Sum of Natural Numbers : 210

**Program-3**

Write a Python Program (using for loop) to find the factorial of a number.

```
num = int(input("Enter a number: "))
factorial = 1
if num < 0:
    print("Sorry, factorial does not exist for negative numbers")
else:
    for i in range(1, num + 1):
        factorial *= i
print("The factorial of", num, "is", factorial)
```

**Output**

Enter a number: 5

The factorial of 5 is 120

## Break and Continue Statement

**Break Statement:** The break statement is used to terminate a loop immediately. It is typically used with conditional statements.

**Continue Statement:** The continue statement skips all the remaining statements in the current iteration of the loop and moves the control to the beginning of the next iteration.

Example :

```
fruits = ['apple', 'banana', 'cherry']
for x in fruits:
    if x == 'banana':
        break
    print(x)
```

Example :

Write a program in Python to check if a number entered by a user is a prime number or not.

```
num = int(input("Enter a number: "))
flag=False
if num > 1:
    for i in range(2, num):
        if (num % i) == 0:
            flag=True
            break
    if flag==True:
        print(num, "is not a prime number")
    else:
        print(num, "is a prime number")
```

Example :

Write a program in Python to print all natural numbers from 1 to 10 except 7.

```
for i in range(1,11):
    if i==7:
        continue
    print(i)
```

Example :

Write a program in Python to print all natural numbers between 1 and 50 (both inclusive) which are not multiple of 3.

```
for x in range(1, 51):
    if x % 3 ==0 :
        continue
    print(x)
```

## Nested loops

Nested loops refers to a loop within a loop.

### Example: Generating a pattern

```
n = int(input("Enter the number of rows: "))
for i in range(n):
    for j in range(i+1):
        print("*", end="")
    print()
```

#### Output

Enter the number of rows: 5

```
*
**
***
****
*****
```

---

## String

**String:** String is a sequence of UNICODE characters. A string can be created by enclosing one or more characters in single, double or triple quotes (' ' or " " or "" ").

### Example

```
>>> str1 = 'Python'
>>> str2 = "Python"
>>> str3 = '''Multi Line
String'''
>>> str4 = """Multi Line
String"""
```

### Terminology

**Whitespace Characters:** Those characters in a string that represent horizontal or vertical space. Example: space (' '), tab ('\t'), and newline ('\n')

## Indexing in Strings

- Indexing is used to access individual characters in a string using a numeric value.
- The index of the first character (from left) in the string is 0 and the last character is n-1
- The index can also be an expression including variables and operators but the expression must evaluate to an integer
- Forward indexing, also known as positive indexing, starts from left to right, with the first character having index 0, second character having index 1, and so on.
- Backward indexing, also known as negative indexing, starts from right to left, with the last character having index -1, second-last character having index -2, and so on

POSITIVE INDEX	0	1	2	3	4	5
STRING	P	Y	T	H	O	N
NEGATIVE INDEX	-6	-5	-4	-3	-2	-1

## Example

<pre>&gt;&gt;&gt; str1 = 'Python' &gt;&gt;&gt; str1[0] 'p'</pre>	<pre>&gt;&gt;&gt; str1 = 'Python' &gt;&gt;&gt; str1[2+3] 'n'</pre>
--	--

## Slicing

Slicing is the process of extracting a substring from a given string. It is done using the operator ':'.  
 Syntax : string[start:end:step].

start : 'start' is the starting index of the substring (inclusive).  
 end : 'end' is the ending index of the substring (exclusive)

step : 'step' is the difference between the index of two consecutive elements. Default value is 1

<b>Case-1</b> <pre>print(str1[:3])</pre> <b>Output</b> PYT	<b>Case-2</b> <pre>print(str1[1:4])</pre> <b>Output</b> YTH	<b>Case-3</b> <pre>print(str1[0:5:2])</pre> <b>Output</b> PTO	<b>Case-4</b> <pre>print(str1[-5:-1])</pre> <b>Output</b> YTHO
<b>Case-5</b> <pre>print(str1[-1:-4])</pre> <b>Output</b> ''	<b>Case-6</b> <pre>print(str1[:-5:-1])</pre> <b>Output</b> NOHT	<b>Case-7</b> <pre>print(str1[:-5])</pre> <b>Output</b> P	

## Mutable Object

If the value of an object can be changed after it is created, It is called mutable object

### Example

Lists, Set and dictionaries.

## Immutable Object

If the value of an object cannot be changed after it is created, it is called immutable object

### Example

Numeric, String and Tuple

## String is Immutable

A string is an immutable data type. It means that the value (content) of a string object cannot be changed after it has been created. An attempt to do this would lead to an error.

### Example

```
>>> str1 = 'Python'
```

```
>>> str1[1]='Y'
```

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

TypeError: 'str' object does not support item assignment

## Traversal of a String

Accessing the individual characters of string i.e. from first

### 1. Using only for Loop

```
str1 = 'Computer'
for ch in str1:
    print(ch, end=' ')
```

### 2. Using For Loop and range function

```
str1 = 'Computer'
for ch in range(0, len(str1)):
    print(str1[ch], end=' ')
```

### 3. Using while Loop

```
str1 = 'Computer'
i = 0
while i < len(str1):
    print(str1[i], end = ' ')
    i+= 1
```

## String Operations

We can perform various operations on string such as concatenation, repetition, membership and slicing.

### Concatenation

Operator : +

```
>>> str1 = "Python"
>>> str2 = "Programming"
>>> str1 + str2
'PythonProgramming'
```

### Repetition

Use : To repeat the given string multiple times.

Repetition operator : \*

```
>>> str1 = "Hello "
>>> str1*5
'Hello Hello Hello Hello Hello'
```

### Membership

Membership is the process of checking whether a particular character or substring is present in a sequence or not. It is done using the 'in' and 'not in' operators.

Example

```
>>> str1 = "Programming in Python"
```

<pre>&gt;&gt;&gt; "Prog" in str1 True</pre>	<pre>&gt;&gt;&gt; "ming in" in str1 True</pre>	<pre>&gt;&gt;&gt; "Pyth " in str1 False</pre>	<pre>&gt;&gt;&gt; "Pyth " not in str1 True</pre>
---	--	---	--

## String Methods/Functions

Python has several built-in functions that allow us to work with strings

<p><b>len()</b> Returns the length of the given string</p> <pre>&gt;&gt;&gt; str1 = 'Hello World!' &gt;&gt;&gt; len(str1) 12</pre>	<p><b>capitalize()</b> converts the first character of a string to capital (uppercase) letter and rest in lowercase.</p> <pre>str1 = "python Programming for 11th" str1.capitalize() 'Python programming for 11th'</pre>
<p><b>title()</b> converts the first letter of every word in the string in uppercase and rest in lowercase</p> <pre>&gt;&gt;&gt; str1 = "python ProGramming for 11th" &gt;&gt;&gt; str1.title() 'Python Programming For 11Th'</pre>	<p><b>lower()</b> Returns the string with all uppercase letters converted to lowercase</p> <pre>&gt;&gt;&gt; str1 = "PYTHON PROGRAMMING for 11th" &gt;&gt;&gt; str1.lower() 'python programming for 11th'</pre>
<p><b>upper()</b> Returns the string with all lowercase letters converted to uppercase</p> <pre>&gt;&gt;&gt; str1 = "python programming for 11th" &gt;&gt;&gt; str1.upper() 'PYTHON PROGRAMMING FOR 11TH'</pre>	<p><b>count()</b> Returns number of times a substring occurs in the given string</p> <pre>&gt;&gt;&gt; str1 = "python programming for 11th" &gt;&gt;&gt; str1.count("p") 2 &gt;&gt;&gt; str1.count("pyth") 1</pre>
<p><b>find()</b> Returns the index of the first occurrence of substring in the given string. If the substring is not found, it returns -1</p> <pre>&gt;&gt;&gt; str1 = "python programming for 11th" &gt;&gt;&gt; str1.find('r') 8 &gt;&gt;&gt; str1.find('u') -1</pre>	<p><b>index()</b> Same as find() but raises an exception if the substring is not present in the given string</p> <pre>&gt;&gt;&gt; str1 = "python programming for 11th" &gt;&gt;&gt; str1.index('r') 8 &gt;&gt;&gt; str1.index('u') Traceback (most recent call last):   File "&lt;stdin&gt;", line 1, in &lt;module&gt; ValueError: substring not found</pre>



<p><b>isalnum()</b></p> <p>The isalnum() method returns True if all characters in the string are alphanumeric (either alphabets or numbers). If not, it returns False.</p> <pre>&gt;&gt;&gt; str1 = 'HelloWorld' &gt;&gt;&gt; str1.isalnum() True &gt;&gt;&gt; str1 = 'HelloWorld2' &gt;&gt;&gt; str1.isalnum() True</pre>	<p><b>isalpha()</b></p> <p>Returns True if all characters in the string are alphabets, Otherwise, It returns False</p> <pre>&gt;&gt;&gt; 'Python'.isalpha() True &gt;&gt;&gt; 'Python 123'.isalpha() False</pre>
<p><b>isdigit()</b></p> <p>returns True if all characters in the string are digits, Otherwise, It returns False</p> <pre>&gt;&gt;&gt; '1234'.isdigit() True &gt;&gt;&gt; '123 567'.isdigit() False</pre>	<p><b>isspace()</b></p> <p>Returns True if has characters and all of them are white spaces (blank, tab, newline)</p> <pre>&gt;&gt;&gt; str1 = '\n\t' &gt;&gt;&gt; str1.isspace() True &gt;&gt;&gt; str1 = 'Hello\n' &gt;&gt;&gt; str1.isspace() False</pre>
<p><b>islower()</b></p> <p>returns True if the string has letters all of them are in lower case and otherwise False.</p> <pre>&gt;&gt;&gt; str1 = 'hello world!' &gt;&gt;&gt; str1.islower() True &gt;&gt;&gt; str1 = 'hello 1234' &gt;&gt;&gt; str1.islower() True &gt;&gt;&gt; str1 = 'hello ??' &gt;&gt;&gt; str1.islower() True</pre>	<p><b>isupper()</b></p> <p>returns True if the string has letters all of them are in upper case and otherwise False.</p> <pre>&gt;&gt;&gt; str1 = 'HELLO WORLD!' &gt;&gt;&gt; str1.isupper() True &gt;&gt;&gt; str1 = 'HELLO 1234' &gt;&gt;&gt; str1.isupper() True &gt;&gt;&gt; str1 = 'HELLO ??' &gt;&gt;&gt; str1.isupper() True</pre>

<p><b>strip()</b></p> <p>Returns the string after removing the whitespaces both on the left and the right of the string</p> <pre>&gt;&gt;&gt; str1 = ' Hello World! ' &gt;&gt;&gt; str1.strip() 'Hello World!'</pre>	<p><b>rstrip()</b></p> <p>Returns the string after removing the whitespaces only on the right of the string</p> <pre>&gt;&gt;&gt; str1 = ' Hello World! ' &gt;&gt;&gt; str1.rstrip() ' Hello World!'</pre>
<p><b>lstrip()</b></p> <p>Returns the string after removing the whitespaces only on the left of the string</p> <pre>&gt;&gt;&gt; str1 = ' Hello World! ' &gt;&gt;&gt; str1.lstrip() 'Hello World! '</pre>	<p><b>replace(oldstr,newstr)</b></p> <p>used to replace a particular substring in a string with another substring</p> <pre>&gt;&gt;&gt; str1 = 'Hello World!' &gt;&gt;&gt; str1.replace('o','*') 'Hell* W*rld!'</pre>
<p><b>partition()</b></p> <p>The partition() function is used to split a string into three parts based on a specified separator.</p> <pre>&gt;&gt;&gt; str1 = 'India is a Great Country' &gt;&gt;&gt; str1.partition('is') ('India ', 'is', ' a Great Country') &gt;&gt;&gt; str1.partition('are') ('India is a Great Country', '', '')</pre>	<p><b>split()</b></p> <p>Returns a list of words delimited by the specified substring. If no delimiter is given then words are separated by space.</p> <pre>&gt;&gt;&gt; str1 = 'India is a Great Country' &gt;&gt;&gt; str1.split() ['India','is','a','Great', 'Country'] &gt;&gt;&gt; str1 = 'India is a Great Country' &gt;&gt;&gt; str1.split('a') ['Indi', ' is ', ' Gre', 't Country']</pre>

### startswith()

startswith() function is used to check whether a given string starts with a particular substring.

### endswith()

endswith() function is used to check whether a given string ends with a particular substring.

```
str = "Python Programming Language"
print(str.startswith("Pyt"))
print(str.startswith("Pyth"))
print(str.endswith("age"))
print(str.endswith("uage"))
print(str.startswith("Pyts"))
```

### Output

```
True
True
True
True
False
```

### join()

str.join(sequence)

returns a string in which the string elements of sequence have been joined by str separator. if few of the members of the sequence are not string, error is generated.

```
>>> '*-*'.join('Python')
'P*-*y*-*t*-*h*-*o*-*n'
>>> '*-*'.join(['Ajay', 'Abhay', 'Alok'])
'Ajay*-*Abhay*-*Alok'
>>> '*-*'.join(['Ajay', 'Abhay', 123])
Traceback (most recent call last):
  File "<stdin>", line 1, in
<module>
TypeError: sequence item 2:
expected str instance, int found
```

**Class: XI**  
**Mid Term Examination**  
**Computer Science (083)**  
**Sample Question Paper (Theory)**

**Time allowed: 3 Hours**  
**70**

**Maximum Marks:**

General Instructions:

Please check this question paper contains 35 questions.

- The paper is divided into 5 Sections- A, B, C, D and E.
- Section A, consists of 18 questions (1 to 18). Each question carries 1 Mark.
- Section B, consists of 7 questions (19 to 25). Each question carries 2 Marks.
- Section C, consists of 5 questions (26 to 30). Each question carries 3 Marks.
- Section D, consists of 2 questions (31 to 32). Each question carries 4 Marks.
- Section E, consists of 3 questions (33 to 35). Each question carries 5 Marks.
- All programming questions are to be answered using Python Language only.

<b>Q No.</b>	<b>Section-A (18 Marks)</b>	<b>Marks</b>
1	Which statement about computer memory is correct? <b>A.</b> RAM is volatile <b>B.</b> ROM is volatile <b>C.</b> Secondary Memory is volatile <b>D.</b> RAM always retains its data	1
2	How much data does one terabyte (TB) represent? <b>A.</b> $2^{10}$ byte <b>B.</b> $2^{10}$ MB <b>C.</b> $2^{10}$ KB <b>D.</b> $2^{10}$ GB	1
3	Convert the decimal number $(345)_{10}$ to Octal: <b>A.</b> 331 <b>B.</b> 431 <b>C.</b> 531 <b>D.</b> 631	1

- 4 If S is a Boolean variable, determine the incorrect Boolean statement for OR laws 1
- A.  $S + 1 = S$
  - B.  $S + 1 = 1$
  - C.  $S + 0 = S$
  - D.  $S + S = S$
- 5 How many bits does ASCII use to represent characters? 1
- A. 5
  - B. 6
  - C. 7
  - D. 8
- 6 Which unit in a computer is responsible for arithmetic and logical operations? 1
- A. Memory
  - B. ALU
  - C. Control Unit
  - D. None of these
- 7 What is the data type of L if  $L = [1, 'Python', 2.5]$ ? 1
- A. List
  - B. String
  - C. Dictionary
  - D. Tuple
- 8 What will be the result of  $343 \% 15$ ? 1
- A. 12
  - B. 13
  - C. 14
  - D. 11
- 9 Which of the following is an invalid variable name? 1
- A. `_rate2`
  - B. `rate2`
  - C. `2_rate`
  - D. `rate_2`
- 10 What is the return type of the `input()` function? 1
- A. list
  - B. integer
  - C. string
  - D. float

- 11 Which of the following is not an assignment operator? 1
- A. /=
  - B. //=
  - C. !=
  - D. +=
- 12 What will be the output of the following code snippet? 1
- ```
X = 'Hi'  
print(len(X) * 2)
```
- A. 2
  - B. 3
  - C. 4
  - D. 6
- 13 Which argument is used with the print() function to specify a different separator? 1
- A. sep
  - B. separator
  - C. end
  - D. tab
- 14 What is the correct precedence order for operators in Python? 1
- A. (), \*\*, \*, +
  - B. \*, +, (), \*\*
  - C. \*, (), +, \*\*
  - D. \*\*, (), \*, +
- 15 How many times will the following code print the word 'Hi'? 1
- ```
for var in range(0, 31, 5):  
    print("Hi")
```
- A. 4
  - B. 5
  - C. 6
  - D. 7
- 16 Which extension is commonly used for Python files? 1
- A. .py
  - B. .p
  - C. .python
  - D. .ppt

- 17 Consider the code given below: 1  
`15=x`  
**ASN** : Above python statement will generate error.  
**RSN**: Variable can be accessed on RHS of assignment operator.  
**A.** Only ASN is true.  
**B.** Only RSN is true.  
**C.** Both ASN and RSN are true, but RSN is not the correct reason for ASN.  
**D.** Both ASN and RSN are true, and R is the correct reason for ASN.
- 18 **ASN**: We can modify list elements. 1  
**RSN**: list is a mutable data type.  
**A.** Only ASN is true.  
**B.** Only RSN is true.  
**C.** Both ASN and RSN are true, but RSN is not the correct reason for ASN.  
**D.** Both ASN and RSN are true, and RSN is the correct reason for ASN.

Q No.	Section-B (14 Marks)	Marks
19	Write down two key differences between a compiler and an interpreter. <div style="text-align: center;"><b>OR</b></div> Expand the following (i.e. write Full Form of): 1. ALU 2. CPU	2
20	Write any two features of Python Programming language.	2
21	Explain the following with a suitable example of each a. Primary Memory b. Secondary Memory <div style="text-align: center;"><b>OR</b></div> Why is RAM called a volatile memory?	1+1=2
22	Write a Python program that accepts distance in meters and converts it into centimeters.	2
23	Write a Python program that prompts user to enter their age. If the age is less than 18, the program should display "You are a minor" otherwise display "You are eligible to vote." <div style="text-align: center;"><b>OR</b></div> What do you mean by error in python? Write down the name of any two types of error.	2

- 24 Define Data-Type and mention any 2 examples. 2  
**OR**  
 Explain Implicit type conversion & Explicit type Conversion
- 25 Consider the python Program given below and write its output 2  
`myStr = "Python Programming"`  
`print(myStr[:2])`  
`print(myStr[-6:-3])`  
`print(myStr[::-1])`  
`print(myStr*2)`

<b>Q No</b>	<b>Section-C (15 Marks)</b>	<b>Marks</b>
26	Explain the following (i) Device Driver with any one example (ii) Assembler (iii) Output Devices with any one example	3
27	Perform the following conversions (a) Convert $(77)_{10}$ to octal. (b) Convert $(232)_8$ to binary. (c) Convert $(74F)_{16}$ to decimal.	3
28	Write a Python program to print the sum of first n odd natural numbers where the value of n is taken as input from the user. <b>OR</b> Write a python program that takes a number as input and prints even or odd accordingly.	3
29	Write a program in python that accepts the radius of a circle and prints its area.	3
30	Write the use of the following function in strings with help of an example (a) len() (b) lower() (c) isupper()	3

<b>Q No.</b>	<b>Section-D (8 Marks)</b>	<b>Marks</b>
--------------	----------------------------	--------------





35

Write a Python program to receive numbers from the user through keyboard until user inputs 0 to end the input process, then the program calculates and displays the sum of given odd numbers and even numbers respectively.

5

example

if the user gives the following sequence of numbers 1 2 3 4 5 6 7 0

Then the output should be as follows:

The Sum of even numbers in given input sequence = 12

The Sum of odd numbers in given input sequence = 16

**OR**

Explain the following with one example of each

- (a) Keyword
- (b) Identifier
- (c) Literal
- (d) Token
- (e) Operator

## LIST IN PYTHON

- List is a **sequence** i.e. sequenced data type.
- List can store multiple values in a single object.
- List is an **ordered sequence** of values/elements.
- Elements of a list are enclosed in **square brackets** [ ], separated by commas.
- List are **mutable** i.e. values in the list can be modified.
- The elements in a list can be of **any type** such as integer, float, string, objects etc.
- List is **heterogeneous** type i.e. collection of different types.

L = [5, 8, 6, 1, 9]

**Length = 5 (Number of elements)**

Positive Indexes	0	1	2	3	4	
	5	8	6	1	9	
	-5	-4	-3	-2	-1	Negative Indexes

### What is a sequence?

Sequence is a positional ordered collection of items, which can be accessed using index. Examples of sequence are List, Tuple, String etc.

### List is Mutable

In Python, lists are mutable. It means that the contents of the list can be changed after it has been created.

```
>>> L = [1, 5, 7, 2]
```

```
>>> L
```

```
[1, 5, 7, 2]
```

Change the second element of list from 5 to 8

```
>>> L[1]=8
```

```
>>> L
```

```
[1, 8, 7, 2]
```

## How to create a list in Python:

### Using integer values

```
>>> L = [1, 5, 7, 2]
```

```
>>> L
```

```
[1, 5, 7, 2]
```

### Using heterogeneous type:

```
>>> L = [1.5, 2, "Hello", 'S']
```

```
>>> L
```

```
[1.5, 2, "Hello", 'S']
```

### Using list() function:

list() function is useful to create or convert any iterable into list. For example, create a list from string.

```
>>> L=list("Hello")
```

```
>>> L
```

```
['H', 'e', 'l', 'l', 'o']
```

### Creating Empty List:

```
>>> L=[]
```

```
>>> L
```

```
[]
```

### Creating empty list using list()

```
>>> L=list()
```

```
>>> L
```

```
[]
```

## Indexing in List:

- Index means the position of an item in an iterable type.
- Indexing can be positive or negative.
- Positive Indexes start from 0 and are positioned like 0, 1, 2, ....
- The index of the first element is always 0 and the last element is n-1 where n is the total number of elements in the list.
- Negative index starts from the last element and is labeled as -1, -2, -3, .....

## For example:

		<b>+ive indexing</b>										
	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>							
<b>L</b>	<b>=</b>	<b>[</b>	<b>5</b>	<b>,</b>	<b>8</b>	<b>,</b>	<b>6</b>	<b>,</b>	<b>1</b>	<b>,</b>	<b>9</b>	<b>]</b>
			<b>-5</b>		<b>-4</b>		<b>-3</b>		<b>-2</b>		<b>-1</b>	
			<b>-ive indexing</b>									

## Accessing elements using index:

### Syntax

Object\_name[Index]

### For Ex:

```
>>> L = [5, 8, 6, 1, 9]
```

```
>>> L[0]
```

```
5
```

```
>>> L[3]
```

```
1
```

```
>>> L[-1]
```

```
9
```

```
>>> L[-4]
```

```
8
```

```
>>> L[5]
```

```
IndexError: list index out of range
```

```
>>> L[-6]
```

**IndexError: list index out of range**

If we try to access the element from outside the index range then the interpreter raises an IndexError.

## LIST OPERATIONS

### Slicing in List:

**Note: Slicing never gives IndexError**

**Syntax:**

**Object\_name[start : stop : interval]**

- Here start, stop and interval all have default values.
- Default value of start is 0, stop is n (if interval is +ive) and -1 if interval is -ive
- Default value interval is 1
- In slicing, start is included and stop is excluded.
- If interval is +ive then slicing is performed from left to right. For Ex:

		<b>+ive indexing</b>				
	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	
L =	[5,	8,	6,	1,	9]	
	<b>-5</b>	<b>-4</b>	<b>-3</b>	<b>-2</b>	<b>-1</b>	
		<b>-ive indexing</b>				

```
>>> L[1:4:1]
```

```
[8, 6, 1]
```

If we give an index out of **range** then it **is** considered **as** the default value.

```
>>> L[1:5:2]
```

```
[8, 1]
```

Start index **is not** given then it consider **as** default value

```
>>> L[:5:2]
```

```
[5, 6, 9]
```

Start, stop **and** interval **is not** given then they consider **as** default value



```
>>> L[-1:2:-1]
[9, 1]
```

## Concatenation of two lists:

**Concatenation is the process of joining the elements of a particular list at the end of another list**

```
>>> L1=[1,2,3]
>>> L2=['a','b','c']
```

**Using concatenation operator (+)**

```
>>> L1+L2
[1, 2, 3, 'a', 'b', 'c']
```

## Repetition Operator

- The repetition operator makes multiple copies of a list and joins them all together. The general format is: list \* n, where 'list' is a list and n is the number of copies to make.

```
>>> L=[0]*3
>>> L
[0, 0, 0]
```

In this example, [0] is a list with one element 0, The repetition operator makes 3 copies of this list and joins them all together into a single list.

```
>>> L = [1, 2, 3]*3
>>> L
[1, 2, 3, 1, 2, 3, 1, 2, 3]
```

In this example, [1, 2, 3] is a list with three elements 1, 2 and 3. The repetition operator makes 3 copies of this list and joins them all together into a single list.

## Membership operator

Membership operator is used to check or validate the membership of an element in the list or sequence.

There are two types of membership operators

- 1. in operator:** It return True if a element with the specified value is present in the List
- 2. not in operator:** It return True if a element with the specified value is not present in the List



## Examples:

```
L = [5,"India", 8, 6, 1]
>>> 5 in L
True

>>> "India" in L
True

>>> 9 in L
False

>>> 9 not in L
True
```

## Traversing a list using loops

Traversing a list means to visit every element of the list one by one. One of the most common methods to traverse a Python list is to use a for loop.

**Ex: Write a program in Python to traverse a list using for loop.**

```
L = [5, 8, 9, 6, 1]
for i in L:
    print(i)
```

**Output:**

```
5
8
9
6
1
```

## Using range() method:

We can traverse a list using range() method

```
L = [5, 8, 9, 6, 1]
n = len(L)
for i in range(n):
    print(L[i])
```

**Output:**

```
5
8
9
```

6  
1

## Built-in functions/methods of list:

### 1: len():

It returns the total number of elements in the list i.e. length of the list.

For ex:

```
>>> L = [8, 6, 3, 9, 1]
>>> len(L)
5
```

It returns the total numbers of elements in the list i.e. 5

### 2: list()

list() is a function used to convert an iterable into a list.

For Ex:

```
>>> S="Python"
>>> L=list(S)
>>> L
['P', 'y', 't', 'h', 'o', 'n']
```

In the above example, a string S is converted into the List.

### 3: append()

appends an element at the end of the list.

#### list.append(item)

append() adds a single element to the end of a list.

The length of the list increases by one.

The method doesn't return any value

```
>>> L = [4,5,9,2,6]
>>> L.append(1)
>>> L
[4, 5, 9, 2, 6, 1]
```

A list is an object. If we append another list onto a list, the parameter list will be a single object at the end of the list.

```
>>> L1=[1,2,3]
>>> L2=[4,5,6]
>>> L1.append(L2)
>>> L1
[1, 2, 3, [4, 5, 6]]
```

#### 4: extend()

extend() is a method of list, which is used to merge two lists.

extend() iterates over its argument and adds each element at the end of the list.

##### **list.extend(iterable)**

Argument of extend() method is any iterable (list, string, set, tuple etc.)

The length of the list increases by a number of elements in its argument.

The method doesn't return any value

```
>>> L1=[1,2,3]
>>> L2=[4,5,6]
>>> L1.extend(L2)
>>> L1
[1, 2, 3, 4, 5, 6]
```

A string is iterable, so if you extend a list with a string, you'll append each character as you iterate over the string.

```
>>> L1=[1,2,3]
>>> S="Python"
>>> L1.extend(S)
>>> L1
[1, 2, 3, 'P', 'y', 't', 'h', 'o', 'n']
```

#### 5: insert()

append() and extend() method is used to add an element and elements of an iterable object respectively at the end of the list. If we want to add an element at any index then we can use insert() method of list.

Syntax: **list\_name.insert(index, element)**

```
>>> L=[1,2,3]
Insert element 8 at index 1 i.e. at 2nd position
>>> L.insert(1,8)
>>> L
[1, 8, 2, 3]
```

#### 6: count()

count() method returns how many times a given element occurs in a List i.e. it returns the frequency of an element in the list.

Syntax: **list\_name.count(element)**

```
>>> L = [1, 2, 8, 9, 2, 6, 2]
>>> L
[1, 2, 8, 9, 2, 6, 2]
>>> L.count(2)
3
```

## 7: index()

index() is a method of the list, which returns the index of the first occurrence of the element.

Syntax: **list\_name.index(element, start, end)**

**element** – The element whose index for first occurrence is to be returned.

**start (Optional)** – The index from where the search begins. Start is included.

**end (Optional)** – The index from where the search ends. End is excluded.

```
>>> L = [5, 8, 9, 6, 1, 8]
>>> L.index(8)
1
>>> L.index(8,2)
5
>>> L.index(8,2,6)
5
>>> L.index(8,2,5)
ValueError: 8 is not in list
>>> L.index(8,2,7)
5
```

## 8: remove()

remove() method of list is used to delete the first occurrence of the given element. If the given element is not exist in the list then it will give the ValueError

```
Syntax: list_name.remove(element)
>>> L
[5, 8, 9, 6, 1, 8]
>>> L.remove(8)
>>> L
[5, 9, 6, 1, 8]
>>> L.remove(4)
ValueError: list.remove(x): x not in list
```

## 9: pop()

Syntax: **list\_name.pop(index)**

Index is optional. If index is not given, pop() method of list is used to remove and return the last element of the list, otherwise remove and return the element of the given index. If Index is out of range then it gives IndexError.

```
>>> L = [5, 8, 9, 6, 1]
>>> L
[5, 8, 9, 6, 1]
#remove the last element from the list
>>> L.pop()
```

```
1
>>> L
[5, 8, 9, 6]
#remove the element from the list whose index is 1
>>> L.pop(1)
8
>>> L
[5, 9, 6]
>>> L.pop(4)
IndexError: pop index out of range
```

### 10: reverse()

reverse() method of the list used to reverse the order of the elements of the list

Syntax: **list\_name.reverse()**

```
>>> L = [5, 8, 9, 6, 1]
>>> L
[5, 8, 9, 6, 1]
>>> L.reverse()
>>> L
[1, 6, 9, 8, 5]
```

### 11: sort()

sort() method can be used to sort List in ascending, descending, or user-defined order. It sort the existing list.

Syntax: **List\_name.sort(reverse=True/False)**

```
>>> L = [5, 8, 9, 6, 1]
>>> L
[5, 8, 9, 6, 1]
#sort the elements of the list in ascending order
>>> L.sort()
>>> L
[1, 5, 6, 8, 9]
#sort the elements of the list in descending order
>>> L.sort(reverse=True)
>>> L
[9, 8, 6, 5, 1]
```

### 12: sorted()

sorted() is a function of list. sorted() function returns a sorted list in ascending order by default. It does not sort or change the existing list.

```
>>> L = [5, 8, 9, 6, 1]
>>> L
[5, 8, 9, 6, 1]
>>> sorted(L)
[1, 5, 6, 8, 9]
>>> L
[5, 8, 9, 6, 1]
#Create a list which contains the element of another list in descending order.
>>> L1=sorted(L, reverse=True)
>>> L1
[9, 8, 6, 5, 1]
```

### 13: min()

min() function of the list is used to find the minimum element from the list.

```
>>> L = [5, 8, 1, 6, 9]
>>> min(L)
1
# If the elements of the list are string then min() returns the minimum element based on the ASCII values.
>>> L=['a','D','c']
>>> min(L)
'D'
```

### 14: max()

max() function of the list is used to find the maximum element from the list.

```
>>> L = [5, 8, 1, 6, 9]
>>> max(L)
9
```

# If the elements of the list are string then max() returns the maximum element based on the ASCII values.

```
>>> L=['a','D','c']
>>> max(L)
'c'
```

### 15: sum()

sum() is a function of list, which returns the sum of all elements of the list.

```
>>> L = [5, 8, 9, 6, 1]
>>> sum(L)
29
```

## Nested List:

A list within another list is called the nested list i.e. list of list.

```
L = [1, [2, 3, 4], 'Python', 3.5]
```

Indices: 0, 1, 2, 3 for the main list; 0, 1, 2 for the nested list; 0-5 for the string 'Python'.

```
>>> L = [1, [2, 3, 4], 'Python', 3.5]
>>> L[1]
[2, 3, 4]
>>> L[1][0]
2
>>> L[2]
'Python'
>>> L[2][2]
't'
```

## Suggested Programs:

**Program-1 :** Write a program in Python to find the maximum and minimum number from the given List.

```
L=[]
c='y'
while c=='y' or c=='Y':
    a=int(input("Enter an integer number to append in the list: "))
    L.append(a)
    c=input("Do you want to add more elements in the list (Y/N): ")
print("List is: \n", L)
print("Minimum(Smallest) number of the list = ",min(L))
print("Maximum(Largest) number of the list = ",max(L))
```

### Output:

```
Enter an integer number to append in the list: 5
Do you want to add more elements in the list (Y/N): Y
Enter an integer number to append in the list: 8
Do you want to add more elements in the list (Y/N): Y
```

Enter an integer number to append in the list: 1  
Do you want to add more elements in the list (Y/N): Y  
Enter an integer number to append in the list: 9  
Do you want to add more elements in the list (Y/N): Y  
Enter an integer number to append in the list: 6  
Do you want to add more elements in the list (Y/N): N  
List is:  
[5, 8, 1, 9, 6]  
Minimum (Smallest) number of the list = 1  
Maximum (Largest) number of the list = 9

**Program-2 : Write a program in Python to find the mean of numeric values stored in the given List.**

```
L=eval(input("Enter a list: "))
print("List is: \n", L)
NL=[ ]
for i in L:
    if isinstance(i, (int, float)) ==True:
        NL.append(i)
print("Numeric List is: \n", NL)
mean=sum(NL)/len(NL)
print("Mean of the numeric values of the list = ",mean)
```

**Output:**

Enter a list: [5, 8, 'hello', (1,2), [7,8], 9.5, 10]  
List is:  
[5, 8, 'hello', (1, 2), [7, 8], 9.5, 10]  
Numeric List is:  
[5, 8, 9.5, 10]  
Mean of the numeric values of the list = 8.125

**Program-3 : Write a program in Python to find the index of the given number from a List using linear search.**

```
L= [10, 20, 30, 40, 50]
print("List is : ",L)
c='Y'
while c=='Y' or c=='y':
    n=int(input("Enter the number to be search in the list: "))
    found=False
    for i in L:
```



```

if i==n:
    print("Element found at index: ",L.index(i))
    found=True
if found==False:
    print("Number is not found in the list")
c=input("Do you want to search more item (Y/N): ")

```

**Output:**

List is : [10, 20, 30, 40, 50]  
Enter the number to be search in the list: 20  
Element found at index: 1  
Do you want to search more item (Y/N): y  
Enter the number to be search in the list: 50  
Element found at index: 4  
Do you want to search more item (Y/N): y  
Enter the number to be search in the list: 15  
Number is not found in the list  
Do you want to search more item (Y/N): n

**Program-4 : Write a program in Python to count the frequency of all elements of a list.**

```

L= [10, 20, 30, 20, 40, 30, 20, 50]
print("List is: ",L)
Uni= []
Freq= []
for i in L:
    if i in Uni:
        index=Uni.index(i)
        Freq[index]+= 1
    else:
        Uni.append(i)
        Freq.append(1)
for i in range(len(Uni)):
    print(Uni[i],":",Freq[i])

```

**Output:**

List is: [10, 20, 30, 20, 40, 30, 20, 50]  
10 : 1  
20 : 3  
30 : 2  
40 : 1  
50 : 1

---

## TUPLE IN PYTHON:

- Tuple is a sequence data type.
- Tuple is a heterogamous collection of data.
- Tuple is an immutable data type i.e. it cannot be changed after being created.
- Tuple is denoted by parenthesis i.e. ( ) and the elements of the tuple are comma separated.

### How to create a tuple in Python:

#Create a tuple with three integer element

```
>>> T= (8, 6, 7)
>>> T
(8, 6, 7)
>>> type(T)
<class 'tuple'>
```

#Create a tuple with a single element. Comma must be used after the element.

```
>>> T= (5,)
>>> T
(5,)
>>> type(T)
<class 'tuple'>
```

#Without parenthesis comma separated values are also treated as tuples.

```
>>> T=5,
>>> T
(5,)
```

#Without a comma, a single element is not tuple.

```
>>> T = (5)
>>> T
5
>>> type(T)
<class 'int'>
```

#Following is string, not tuple

```
>>> T= ('Hello')
```



## Tuple Operations

### Concatenation

Using + operator

```
>>> T1=(1,2,3)
>>> T2=(4,5,6)
>>> T=T1+T2
>>> T
(1, 2, 3, 4, 5, 6)
```

### Repetition

Tuple can be created using the repetition operator, \*.

The repetition operator makes multiple copies of a tuple and joins them all together. The general format is: T \* n,

Where T is a tuple and n is the number of copies to make.

```
>>> T= (1, 2)*3
>>> T
(1, 2, 1, 2, 1, 2)
```

In this example, (1, 2) is a tuple with two elements 1 and 2, The repetition operator makes 3 copies of this tuple and joins them all together into a single tuple.

### Membership

Membership operator is used to check or validate the membership of an element in the tuple or sequence.

There are two types of membership operators

1. in operator: It return True if a element with the specified value is present in the tuple
2. not in operator: It returns True if an element with the specified value is not present in the tuple.

Examples:

```
T = (5,"India", 8, 6, 1)
>>> 5 in T
True
>>> "India" in T
True
>>> 9 in T
False
>>> 9 not in T
True
```

## Slicing in Tuple:

Note: Slicing never give IndexError

Syntax:

**Object\_name[start : stop : interval]**

- Here start, stop and interval all have default values.
- Default value of start is 0, stop is n (if interval is +ive) and -1 if interval is -ive
- Default value interval is 1
- In slicing start is included and stop is excluded.
- If interval is +ive then slicing is performed from left to right. For Ex:

		+ive indexing			
	0	1	2	3	4
T =	( 5 ,	8 ,	6 ,	1 ,	9 )
	-5	-4	-3	-2	-1
		-ive indexing			

```
>>> T[1:4:1]
(8, 6, 1)
```

If we give an index out of range then it is considered as the default value.

```
>>> T[1:5:2]
(8, 1)
```

If start index is not given then it consider as default value

```
>>> T[:5:2]
(5, 6, 9)
```

If start >= stop and interval is positive the slicing will return empty list

```
>>> T[4:2]
()
```

If interval is negative slicing performed from right to left

Access the list element where start index is 4 and stop index is 1 in reverse order

```
>>> T[4:1:-1]
(9, 1, 6)
```

Access the list element where start index is 5 and stop index is 0 with interval 2 in reverse order

```
>>> T[5:0:-2]
(9, 6)
```

```
>>>T[-1:-4:-1]
(9, 1, 6)
>>>T[-1:-4:-1]
(9, 1, 6)
```

## Built-in functions/methods of tuple:

### 1: len():

len() is a function of tuple.

It returns the total number of elements in the tuple i.e. length of the tuple.

For ex:

```
>>> T = (8, 6, 'Hello', 9, 1)
>>> len(T)
5
```

It returns the total numbers of elements in the tuple i.e. 5

### 2: tuple()

tuple() is a function used to convert an object into tuple.

For Ex:

```
>>> S="Python"
>>> T=tuple(S)
>>> T
('P', 'y', 't', 'h', 'o', 'n')
```

In the above example, a string S is converted into the tuple.

### 3: count()

count() is a function used to count() the number of occurrences of an element in the tuple.

```
>>> T = (1, 2, 3, 2, 4, 2, 5)
>>> T.count(2)
3
```

### 4: index()

It returns the index of the first occurrence of the element

```
>>> T = (1, 2, 3, 2, 4, 2, 5)
>>> T.index(2)
1
```

### 5: sorted()

It is a built-in function to sort the elements of Tuple.

It return a new list of sorted elements of the tuple in ascending order (by default)  
It does not change the existing tuple.

```
T = (1, 2, 3, 2, 4, 2, 5)
print(sorted(T))
Output:
[1, 2, 2, 2, 3, 4, 5]
```

For sort the elements in descending order use reverse = True

```
T = (1, 2, 3, 2, 4, 2, 5)
print(sorted(T, reverse=True))
Output:
[5, 4, 3, 2, 2, 2, 1]
```

#### 6: min()

It is a built-in function of the tuple. It returns the smallest element from the tuple.

```
T = (5, 8, 3, 9, 6, 4)
print(min(T))
Output:
3
```

#### 7: max()

It is a built-in function of the tuple. It returns the largest element from the tuple.

```
T = (5, 8, 3, 9, 6, 4)
print(max(T))
Output:
9
```

#### 8: sum()

It is a built-in function of the tuple. It returns the sum of all elements of the tuple.

```
T = (5, 8, 3, 9, 6, 4)
print(sum(T))
Output:
35
```

#### Nested Tuple

A nested tuple is a tuple that has been placed inside of another tuple.

```
>>> T = (5, (8, 6, 4), 3, 9)
```

```

>>> T
(5, (8, 6, 4), 3, 9)
>>> T[1]
(8, 6, 4)
>>> T[1][1]
6

```

A list can be an element of a tuple.

```

>>> T = (5, [8, 6, 4], 3, 9)
>>> T
(5, [8, 6, 4], 3, 9)

>>> T[1][2]
4

```

List is a mutable type. So we can change the value of the list, even if it is nested inside the tuple.

```

>>> T[1][2]=7
>>> T
(5, [8, 6, 7], 3, 9)

```

### Suggested Programs:

**Program-1 :** Write a program in Python to find the maximum and minimum number from the given tuple.

```

T= (5, 8, 9, 6, 2, 5)
print("Tuple is: \n", T)
print("Minimum(Smallest) number of the Tuple = ",min(T))
print("Maximum(Largest) number of the Tuple = ",max(T))

```

#### Output:

```

Tuple is:
(5, 8, 9, 6, 2, 5)
Minimum(Smallest) number of the Tuple = 2
Maximum(Largest) number of the Tuple = 9

```

**Program-2 :** Write a program in Python to find the mean of numeric values stored in the given tuple.



```
T=eval(input("Enter a numeric tuple: "))
print("Tuple is: \n", T)
mean=sum(T)/len(T)
print("Mean of the numeric values of the tuple = ",mean)
```

### Output:

Enter a numeric tuple: (2, 8, 9, 6, 5)

Tuple is:

(2, 8, 9, 6, 5)

Mean of the numeric values of the tuple = 6.0

**Program-3 : Write a program in Python to find the index of the given number from a tuple using linear search.**

```
T= (5, 8, 9, 2, 6)
print("Tuple is : ",T)
c='Y'
while c=='Y' or c=='y':
    n=int(input("Enter the number to be search in the tuple: "))
    found=False
    for i in T:
        if i==n:
            print("Element found at index: ",T.index(i))
            found=True
    if found==False:
        print("Number is not found in the list")
    c=input("Do you want to search more item (Y/N): ")
```

### Output:

Tuple is : (5, 8, 9, 2, 6)

Enter the number to be search in the tuple: 8

Element found at index: 1

Do you want to search more item (Y/N): y

Enter the number to be search in the tuple: 2

Element found at index: 3

Do you want to search more item (Y/N): n

**Program-4 : Write a program in Python to count the frequency of all elements of a tuple.**

```
T= (10, 20, 30, 20, 40, 30, 20, 50)
print("Tuple is: ",T)
Uni= ()
for i in T:
    if i not in Uni:
        Uni = Uni + (i,)
for i in Uni:
    print(i,":",T.count(i))
```

**Output:**

Tuple is: (10, 20, 30, 20, 40, 30, 20, 50)

10 : 1

20 : 3

30 : 2

40 : 1

50 : 1

---

# DICTIONARY IN PYTHON

## Introduction to Dictionary:

- Dictionary is a sequence data type.
- Dictionary is a mutable data type i.e. it can be changed after being created.
- Dictionary is denoted by curly braces i.e. {}
- Dictionaries are used to store data values in key : value pairs
- Keys of the dictionary are unique.

```
>>> D = {1:10, 2:20, 3:30}
>>> D
{1: 10, 2: 20, 3: 30}
```

## Accessing items in a dictionary using keys:

```
>>> D = {1:10, 2:20, 3:30}
>>> D
{1: 10, 2: 20, 3: 30}
```

We can access the values of any key from the dictionary using following syntax:

Dict\_Name[key\_name]

```
>>> D[1]
10
>>> D[2]
20
>>> D[3]
30
```

If the key is duplicated at the time of dictionary creation then the value assigned at the last will be stored in the key.

```
>>> D = {1:10, 2:20, 3:30, 1:40, 4:50}
>>> D
{1: 40, 2: 20, 3: 30, 4: 50}
>>> D = {'A':10, 'B':20, 'C':30}
>>> D.get('A')
10
```

## Mutability of a dictionary:

Mutability means the values of a key can be updated in the dictionary.

Dict\_Name[Key\_Name] = New\_Value

```
>>> D = {'A':10,'B':20,'C':30}
Change the value of key 'A' from 10 to 15.
>>> D['A']=15
>>> D
{'A': 15, 'B': 20, 'C': 30}
```

### Adding a new item in dictionary:

Adding an item to the dictionary is done by using a new index key and assigning a value to it:  
Dict\_Name[New\_Key\_Name] = Value

```
>>> D = {1:10, 2:20, 3:30}
>>> D
{1: 10, 2: 20, 3: 30}
>>> D[4]=40
>>> D
{1: 10, 2: 20, 3: 30, 4: 40}
```

### Traversing a dictionary

- You can loop through a dictionary by using for loop.
- When looping through a dictionary, the return values are the keys of the dictionary, but there are methods to return the values as well.

We can use following methods to traversing a dictionary:

#### keys()

It returns the keys of the dictionary, as a list.

```
>>> D = {1:10, 2:20, 3:30}
>>> D.keys()
dict_keys([1, 2, 3])
#Print the keys of the dictionary.
D = {1:10, 2:20, 3:30}
for i in D.keys():
    print(i)
```

Output:

```
1
2
3
```

#### values()

It returns the values of the dictionary, as a list.

```
>>> D = {1:10, 2:20, 3:30}
>>> D.values()
```

```
dict_values([10, 20, 30])

#Print the values of the dictionary.
D = {1:10, 2:20, 3:30}
for i in D.values():
    print(i)
```

#### Output:

```
10
20
30
```

#### items()

It returns the key-value pairs of the dictionary, as tuples in a list.

```
>>> D = {1:10, 2:20, 3:30}
>>> D.values()
dict_items([(1, 10), (2, 20), (3, 30)])

#Print all keys and values of the dictionary.
D = {1:10, 2:20, 3:30}
for i in D.items():
    print(i)
```

#### Output:

```
(1, 10)
(2, 20)
(3, 30)
```

### Built-in functions/methods of dictionary:

#### 1. len()

It returns the number of keys in the list.

```
>>> D = {1:10, 2:20, 3:30}
>>> len(D)
3
```

#### 2. dict()

The dict() function is used to create a dictionary.

```
>>> D = dict(name = "Shiva", age = 26, country = "India")
>>> D
{'name': 'Shiva', 'age': 26, 'country': 'India'}
```

### 3. get()

The get() method returns the value of the specified key.

```
>>> D = {1:10, 2:20, 3:30}
#Return the value of key '1'
>>> D.get(1)
10
#Returns the value of key '3'
>>> D.get(3)
30
```

### 4. update()

The update() method inserts specified elements to the dictionary or merge two dictionaries.  
dictionary.update(iterable)

```
>>> D = {1:10, 2:20, 3:30}
>>> D.update({4:40})
>>> D
{1: 10, 2: 20, 3: 30, 4: 40}
```

If key is already exist in the dictionary then the value of the existing key will be update:

```
>>> D.update({3:35})
>>> D
{1: 10, 2: 20, 3: 35, 4: 40}
```

Merge two dictionaries using update() method

```
>>> D1={1:10,2:20}
>>> D1
{1: 10, 2: 20}
>>> D2={3:30,4:45}
>>> D2
{3: 30, 4: 45}
>>> D1.update(D2)
>>> D1
{1: 10, 2: 20, 3: 30, 4: 45}
```

### 5. del

del is a keyword.

del is used to delete an element from the dictionary using keyname.

```
>>> D = {1:10, 2:20, 3:30}
>>> D
```

```
{1: 10, 2: 20, 3: 30}
>>> del D[1]
>>> D
{2: 20, 3: 30}
```

del can also be used to delete the complete dictionary

```
>>> del D
>>> D
NameError: name 'D' is not defined.
```

### 6. clear()

The clear() method is used to delete all the elements (key:value pairs) from a dictionary. It does not delete the structure of the dictionary

```
>>> D = {1:10, 2:20, 3:30}
>>> D
{1: 10, 2: 20, 3: 30}
>>> D.clear()
>>> D
{}
```

### 7. pop()

pop() method is used to delete a specific element from the dictionary and also return the value of the deleted element.

```
>>> D = {1:10, 2:20, 3:30}
>>> D
{1: 10, 2: 20, 3: 30}
>>> D.pop(2)
20
>>> D
{1: 10, 3: 30}
```

### 8. popitem()

popitem() method is used to delete the last element from the dictionary. It return the (key, value) pair of deleted elements in tuple type.

```
>>> D = {1:10, 2:20, 3:30}
>>> D
{1: 10, 2: 20, 3: 30}
>>> D.popitem()
(3, 30)
```

## 9. setdefault()

setdefault() method returns the value of the specified key. If the key does not exist in the dictionary it insert the key into dictionary, with the specified value

```
>>> D = {1:10, 2:20, 3:30}
Return the value of key '2'
>>> D.setdefault(2)
20
```

Return the value of key '3'. If key is already exists in the dictionary, 2nd argument does not effects

```
>>> D.setdefault(3,50)
30
>>> D
{1: 10, 2: 20, 3: 30}
```

Return the value of key '4'. If the key is not found in the dictionary, then add the key and specified value and then return the value of the key.

```
>>> D.setdefault(4,40)
40
>>> D
{1: 10, 2: 20, 3: 30, 4: 40}
```

If the key is not found in the dictionary, then add the key and None as the value of key (if value is not passed in the argument). It will not return anything.

```
>>> D.setdefault(5)
>>> D
{1: 10, 2: 20, 3: 30, 4: 40, 5: None}
```

## 10. max()

max() function returns the largest key from the dictionary. Comparison done on the bases of ASCII values

```
>>> D={1: 10, 2: 20, 3: 30, 4: 40}
>>> max(D)
4
```

If keys are string, here 'h' has the largest ASCII value.

```
>>> D={'a':10, 'B':20, 'h':30}
>>> max(D)
```



```
'h'
```

### 11. min()

min() function returns the smallest key from the dictionary.

```
>>> D={1: 10, 2: 20, 3: 30, 4: 40}
>>> min(D)
1
```

If keys are string, here 'B' has the largest ASCII value.

```
>>> D={'a':10,'B':20,'h':30}
>>> min(D)
'B'
```

### 12. fromkeys()

The fromkeys() method creates and returns a dictionary. It is useful when we want to create a dictionary with multiple keys which have the same value.

Syntax: dict.fromkeys(keys, value)

```
>>> K=('A','B','C')
>>> V=10
>>> D=dict.fromkeys(K,V)
>>> D
{'A': 10, 'B': 10, 'C': 10}
```

### 13. copy()

The copy() method returns a copy of the specified dictionary.

dictionary.copy()

```
>>> D={1:10,2:20}
Create a copy of dictionary D and store in D1
>>> D1=D.copy()
>>> D1
{1: 10, 2: 20}
```

### 14. sorted()

sorted() function sort the dictionary keys in ascending order

```
>>> D={1:10,4:40,3:30,2:20}
Return the keys of dictionary in sorted order
>>> sorted(D)
[1, 2, 3, 4]
Return the keys of dictionary in sorted order
>>> sorted(D.keys())
[1, 2, 3, 4]
```

Return the values of dictionary **in sorted** order

```
>>> sorted(D.values())
```

```
[10, 20, 30, 40]
```

Return the (key, value) pairs of dictionary **in sorted** order

```
>>> sorted(D.items())
```

```
[(1, 10), (2, 20), (3, 30), (4, 40)]
```

### Suggested Programs:

Program-1 : Write a Program in Python to count the number of times a character appears in a given string using a dictionary.

```
S=input("Enter a string: ")
D={}
for c in S:
    if c in D:
        D[c]+=1
    else:
        D[c]=1
print("Frequency of Characters in String: \n",D)
```

**Output:**

Enter a string: S P SHARMA

Frequency of Characters in String:

```
{'S': 2, ' ': 2, 'P': 1, 'H': 1, 'A': 2, 'R': 1, 'M': 1}
```

Program-2 : Write a Program in Python to create a dictionary with names of employees, their salary and access them

```
Emp = { }
while True :
    name = input("Enter employee name : ")
    salary = int(input("Enter employee salary : "))
    Emp[name] = salary
    c = input("Do you want to add more employee details(y/n) :")
    if c not in "yY" :
        break
print(Emp)
```

**Output:**

Enter employee name : Amit

Enter employee salary : 20000

Do you want to add more employee details(y/n) :y

Enter employee name : Sachin

Enter employee salary : 25000

Do you want to add more employee details(y/n) :y

Enter employee name : Saanvi

Enter employee salary : 28000

Do you want to add more employee details(y/n) :n

{'Amit': 20000, 'Sachin': 25000, 'Saanvi': 28000}

# MODULES IN PYTHON

Introduction to Modules:

- A Python file which is used in another file is called a Python module.
- A module can contain Python code, definitions of functions, set of functions or classes.
- The extension of the Python module file is also .py
- Common modules used in Python are:
  - math module
  - random module
  - statistics module

## Importing module

There are different ways by which we can import a module

### 1. import statement

'import' statement can be used to import a module. It provides access to all attributes (like variables, constants etc.) and methods or functions of the module.

**import <module\_name>**

For example: **import math**

To access multiple modules, we can use following syntax

**import module1, module2, ....., moduleN**

For example: **import math, random**

To access particular methods from any module, use following syntax

**import <module\_name>.<function\_name>**

For example: **import random.randint(3,6)**

### 2. from statement

'from' statement is also used to import specific attributes or objects from a module.

**from <module\_name> import <function\_name>**

or

**from <module\_name> import function1, function2, .....**

For example:

**from math import sqrt**

or

**from math import sqrt, floor**

### 3. import \*

**import \*** statement used to import all objects from a module.

**from <module\_name> import \***

For example:

```
from math import *
```

### Module aliasing:

You can create an alias (nickname) of a module using 'as' keyword.

```
import <module_name> as <alias_name>
```

For example:

```
import random as r
```

### Attributes, functions and methods of math module:

#### 1. pi

It is a mathematical constant. It returns the value of pi

```
>>> import math  
>>> math.pi  
3.141592653589793
```

#### 2. e

It is a mathematical constant that return the value of e, where  $e = 2.718281$ , It is also called Euler's number

```
>>> import math  
>>> math.e  
2.718281828459045
```

#### 3. sqrt()

It returns the square root of a number.

Syntax: **math.sqrt(number)**

```
>>> import math  
>>> math.sqrt(25)  
5.0  
>>>math.sqrt(50)  
7.0710678118654755
```

#### 4. ceil()

Syntax: **math.ceil(x)**

It returns the smallest integer that is greater than or equal to x.

```
>>> import math  
>>> math.ceil(4.5)  
5
```

```
>>> math.ceil(6.0)
6
>>> math.ceil(7.1)
8
```

## 5. floor()

Syntax: `math.floor(x)`

It returns the largest integer that is less than or equal to x.

```
>>> import math
>>> math.floor(4.5)
4
>>> math.floor(6.0)
6
>>> math.floor(7.1)
7
```

## 6. pow()

Syntax: `math.pow(x,y)`

It returns the value of xy

```
>>> import math
>>> math.pow(3,2)
9.0
>>> math.pow(5,0)
1.0
>>> math.pow(10,-1)
0.1
```

## 7. fabs()

Syntax: `math.fabs(x)`

It returns the absolute value of x i.e. the value without sign. Its return type is float.

```
>>> import math
>>> math.fabs(-10)
10.0
>>> math.fabs(15)
15.0
>>> math.fabs(0)
0.0
```

## 8. sin()

Syntax: `math.sin(x)`

It returns the sine of x in radians

Angle in radian = Angle in degree x ( $\pi/180$ )

Where  $\pi = 22/7$  or 3.14

**Convert 30° in radians**

30° in radians = Angle in degree x ( $\pi/180$ )

= 30 \* 3.14 /180

=0.524 (approx)

**Convert 60° in radians**

60° in radians = Angle in degree x ( $\pi/180$ )

= 60 \* 3.14 /180

=1.047 (approx)

```
>>> import math
```

```
>>> math.sin(0)
```

```
0.0
```

```
>>> math.sin(0.524)
```

```
0.5003474302699141
```

```
>>> math.sin(1.047)
```

```
0.8659266112878228
```

## 9. cos()

Syntax: `math.cos(x)`

It returns the cosine of x in radians

```
>>> import math
```

```
>>> math.cos(0)
```

```
1.0
```

```
>>> math.cos(0.524)
```

```
0.8658247218821448
```

```
>>> math.cos(1.047)
```

```
0.5001710745970701
```

## 10. tan()

Syntax: `math.tan(x)`

It returns the tangent of x in radians

```
>>> import math
```

```
>>> math.tan(0)
```

```
0.0
```

```
>>> math.tan(0.524)
```

```
0.5778853590392409
```

```
>>> math.tan(0.524)
```

```
0.5778853590392409
```

## Attributes, functions and methods of random module:

random module is used to generate random numbers, which are used to generate captcha code, in computer games like throwing of a dice, picking a number or flipping a coin, shuffling cards, creating lottery scratch cards. Random numbers are also used in online quizzes, tests to shuffle questions etc. To use random module we need to import the random module.

### random()

Syntax: `random.random()`

random() method generates a random number from 0 to 1. It does not takes any parameter and returns floating point values between 0 and 1 (including 0, but excluding 1)

```
>>> import random as r
>>> r.random()
0.13939737870230484
>>> r.random()
0.5121394221754262
>>> r.random()
0.5622319595776127
```

### randint()

Syntax: `random.randint(a, b)`

randint() method accepts two parameters and returns a random integer number between a and b (both a and b are inclusive). It can generate duplicate numbers.

```
>>> import random as r
>>> r.randint(4,9)
4
>>> r.randint(4,9)
5
>>> r.randint(4,9)
7
```

### randrange()

Syntax: `random.randrange(start, stop, step)`

randrange() method generates a random integer number between start and stop, where start is inclusive and stop is exclusive. Here start and step are optional parameters. The default value of start is 0 and step is 1.

```
>>> import random as r
#Generate any random number between 0 to 9 (0 include, but 9 exclude)
>>>r.randrange(9)
```



```
2
#Generate any random number between 4 to 9 (4 include, but 9 exclude)
>>> r.randrange(4,9)
5
#Generate any even random number between 3 to 9 (3 include, but 9 exclude)
>>>r.randrange(3,9,2)
4
```

## Attributes, functions and methods of statistics module

The statistics module implements many statistical methods like mean(), median() and mode etc. To use statistical methods we need to import the statistics module.

import statistics as s

### mean()

Syntax: **statistics.mean(data)**

The mean() method returns the arithmetic mean (average) of the given data set.

### median()

Syntax: **statistics.median(data)**

The mean() method calculates the median (middle value) of the given data set. This method also sorts the data in ascending order before calculating the median.

Median =  $\{(n + 1) / 2\}$ th value, where n is the number of values in a set of data. In order to calculate the median, the data must first be sorted in ascending order. The median is the number in the middle.

**Note:** If the number of data values is odd, it returns the exact middle value. If the number of data values is even, it returns the average of the two middle values.

### mode()

Syntax: **statistics.mode(data)**

The mean() method calculates the mode (central tendency) of the given data set.

```
import statistics as s
L = [1, 4, 5, 7, 4, 7, 4, 10, 12]
print("Mean of the data Set: ", s.mean(L))
print("Median of the data Set: ", s.median(L))
print("Mode of the data Set: ", s.mode(L))
```

#### Output:

```
Mean of the data Set: 6
Median of the data Set: 5
Mode of the data Set: 4
```

## Multiple choice questions

1. What are the basic steps in problem solving?

- A) Execution, Debugging, Testing
- B) Analyzing the problem, developing an algorithm, coding, testing and debugging
- C) coding, analysis, testing, debugging
- D) debugging, coding, testing, analyzing

**Answer: B) Analyzing the problem, developing an algorithm, coding, testing and debugging**

2. Which of the following is a representation of an algorithm using graphical symbols?

- A) pseudo code
- B) Flowchart
- C) Python code
- D) variable

**Answer: B) Flowchart**

3. What is the purpose of comments in Python code?

- A) To add logic to the program
- B) making the code more confusing
- C) to explain the code to other programmers
- D) To hide the code from others

**Answer: C) to explain the code to other programmers**

4. Which data type is used to store a single character in Python?

- A) integer
- B) string
- C) character
- D) four

**Answer: B) string**

5. Which operator checks whether an element is present in a sequence or not?

- A) membership operator
- B) identity operator
- C) assignment operator
- D) relational operator

**Answer: A) membership operator**

6. What is the purpose of break statement in a loop?

- A) to continue to the next iteration of the loop
- B) premature exit from the loop
- C) leaving the loop completely

D) repeating the loop indefinitely

**Answer: B) premature exit from the loop**

**7. Which built-in function is used to find the length of a string in Python?**

A) size()

B) length()

C) count()

D) len()

**Answer: D) len()**

**8. What is the output of the following Python code?**

```
x = 5
```

```
you = 2
```

```
result = x/y
```

```
print(result)
```

A) 7

B) 2.5

C) 2

D) error

**Answer: B) 2.5**

**9. Which of the following is an example of mutable data type in Python?**

A) string

B) integer

C) list

D) tuple

**Answer: C) list**

**10. What is the purpose of the range() function in the for loop in Python?**

A) Generating a sequence of numbers

B) performing mathematical operations

C) Checking membership in a list

D) Calculating the factorial of a number

**Answer: A) Generating a sequence of numbers**

**11. Which of the following is not a valid Python variable name?**

A) my\_variable

B) 123\_variable

C) \_variable

D) Variable123

**Answer: B) 123\_variable**

**12. Which operator is used to check whether two values are equal in Python?**

- A) ==
- B) !=
- C) <=
- D) >

**Answer: A) ==**

**13. What is the result of the expression 5%2 in Python?**

- A) 2
- B) 2.5
- C) 0.5
- D) 1

**Answer: D) 1**

**14. What does the strip() method do for a string in Python?**

- A) Removes all the vowels from the string
- B) Removes leading and trailing spaces from the string
- C) Converts the string to uppercase
- D) splits the string into a list

**Answer: B) Removes leading and trailing spaces from the string**

**15. Which of the following is a logical operator in Python?**

- A) +
- B) %
- C) and
- D) >

**Answer: C) and**

**16. What does the input() function do in Python?**

- A) displays the output on the screen
- B) accepts input from the user
- C) performs mathematical calculations
- D) converts data types

**Answer: B) accepts input from the user**

**17. Which of the following statements will create a list?**

- A) L = list()
- B) L = []
- C) L = list([1, 2, 3])

D) All of the above

**Answer: D) All of the above**

**18. What is the output when we execute a list('hello')?**

A) ['h', 'e', 'l', 'l', 'o']

B) ['hello']

C) ['llo']

D) ['olleh']

**Answer: A) ['h', 'e', 'l', 'l', 'o']**

**19. Suppose L is [5, 3, 9, 4], what is max(L)?**

A) 5

B) 3

C) 9

D) 4

**Answer: C) 9**

**20. Suppose L is [5, 3, 9, 4], what is min(L)?**

A) 5

B) 3

C) 9

D) 4

**Answer: B) 3**

**21. Suppose L is [5, 3, 9, 4], what is sum(L)?**

A) 5

B) 3

C) 9

D) 21

**Answer: D) 21**

**22. Suppose L is [2, 3, 8, 4, 5], What is L[-1]?**

A) 2

B) 5

C) 8

D) 4

**Answer: B) 5**

**23. Suppose L is [2, 3, 8, 4, 5], What is L[:-1]?**

A) [2, 3, 8, 4, 5]

B) [2, 3, 8, 4]

C) [5, 4, 8, 3, 2]

D) [4, 8, 3, 2]

**Answer: B) [2, 3, 8, 4]**

**24. Which of the following is a Python tuple?**

A) [1, 2, 3]

B) (1, 2, 3)

C) {1, 2, 3}

D) {}

**Answer: B) (1, 2, 3)**

**25. If T = (1, 2, 4, 3), which of the following is incorrect?**

A) print(T[3])

B) T[3] = 45

C) print(max(T))

D) print(len(T))

**Answer: B) T[3] = 45**

**26. What will be the output of the following Python code?**

```
>>>T = (1, 2)
>>>2 * T
```

A) (1, 2, 1, 2)

B) [1, 2, 1, 2]

C) (1, 1, 2, 2)

D) [1, 1, 2, 2]

**Answer: C) (1, 2, 1, 2)**

**27. Write the output of the following code?**

```
>>> L1=[1,55,100,6,453,2,66,23,56]
>>> L2=sorted(L1)
>>> L2
```

A) [1, 2, 6, 23, 55, 56, 66, 100, 453]

B) [1, 55, 100, 6, 453, 2, 66, 23, 56]

C) []

D) Error

**Answer: A) [1, 2, 6, 23, 55, 56, 66, 100, 453]**

**28. Which method deletes the last inserted (key, value) pair from the dictionary?**

A) pop()

B) popitem()

- C) del()
- D) del

**Answer: B) popitem()**

**29. \_\_\_\_\_ is a mapping of unique keys to values. It consists of key-values pairs.**

- A) List
- B) Dictionary
- C) String
- D) Tuple

**Answer: B) Dictionary**

**30. Write output of the following code in interactive mode/shell?**

```
>>> dict1={'Student': (1, 'Sachin', 'Computer'), 'Teacher': ('Hindi', 1001, 'Saanvi')}
>>> dict1["Teacher"][2]
```

- A) 'Sachin'
- B) 'Hindi'
- C) 'Saanvi'
- D) 'Teacher'

**Answer: C) 'Saanvi'**

**31. To include the use of functions which are present in the random library, we must use the option:**

- A) import random
- B) random.h
- C) import.random
- D) random.random

**Answer: A) import random**

**32. What can be the output of the following Python code?**

```
random.randrange(1,100,10)
```

- A) 32
- B) 67
- C) 91
- D) 80

**Answer: D) 80**

### Short Answer Questions

**1. Define the term algorithm.**

Answer: An algorithm is a step-by-step procedure for performing a specific task.

2. **What are the three main types of errors in programming, and briefly explain each type?**

**Answer:**

Syntax errors : errors in the code's structure

Logical errors : errors in the program's logic

Runtime errors : errors that occur during program execution

3. **Explain the purpose of the break and continue statements in Python.**

**Answer:**

The break statement is used to exit a loop prematurely while the continue statement is used to skip the remaining code in the current iteration of a loop and move on to the next iteration.

4. **Describe the difference between a mutable and an immutable data type in Python, providing an example of each.**

**Answer:**

Mutable data types can be changed or modified after creation. Example List

Immutable data types cannot be changed once created. Example tuple.

5. **Explain the concept of "type conversion" in Python.**

**Answer:**

Type conversion refers to the process of changing the data type of a value from one type to another. In Python, this can be done explicitly or implicitly.

6. **Explain the role of data types in Python.**

**Answer:**

Data types define the kind of data a variable can hold. Examples of data types include int, float, strings lists etc.

7. **What is the difference between List and Tuple?**

- List is a mutable type whereas Tuple is an immutable type.
- List is denoted by square brackets i.e. [ ] whereas Tuple is denoted by parenthesis i.e. ( )

8. **What is the expected output of the following Python code.**

```
import random
M=[5, 10, 15, 20, 25, 30]
for i in range(1,3):
F=random.randint(2,5) - 1
S=random.randint(3,6) - 2
T=random.randint(1,4)
print(M[F], M[S], M[T], sep="#")
```



- A) 10#25#15  
20#25#25
- B) 5#25#20  
25#20#15
- C) 30#20#20  
20#25#25
- D) 10#15#25#  
15#20#10#

**Answer:**

- A) 10#25#15  
20#25#25

9. Write the output of the following Python code.

```
A = ['1']  
A.extend('234')  
print(A)
```

Output: ['1', '2', '3', '4']

10. Write the output of the following Python code.

```
a=[2,5,3,4]  
a[2:2] =[2]  
print(a)
```

Output: [2, 5, 2, 3, 4]

11. Write the output of the following Python code.

```
T=(10,20,30,(10,20,30),40)  
print(T.index(20))
```

Output: 1

12. Write the output of the following Python code.

```
T = ("Akshay", "Prashant", "Anirudh", "Sachin", "Amit", "Yamini")
print(T[-1:-4:-1])
```

Output: ('Yamini', 'Amit', 'Sachin')

13. Write a Python program to check whether a dictionary is empty or not.

```
d = eval(input("Input a dictionary: "))
if len(d) == 0:
    print("The given dictionary is empty")
else:
    print("The given dictionary is not empty")
```

Output:

```
Input a dictionary: {}
The given dictionary is empty
Input a dictionary: {1:10}
The given dictionary is not empty
```

14. Write a Python program to merge two dictionaries.

```
D1 = {1: 'a', 2: 'b'}
D2 = {2: 'c', 4: 'd'}
print("Using | Operator")
print(D1 | D2)
print("Using ** Operator")
print(**D1, **D2)
print("Using copy() and update()")
D3 = D1.copy()
D3.update(D2)
print(D3)
```

Output:

```
Using | Operator
{1: 'a', 2: 'c', 4: 'd'}
Using ** Operator
{1: 'a', 2: 'c', 4: 'd'}
Using copy() and update()
{1: 'a', 2: 'c', 4: 'd'}
```

15. Write the output of the following Python code.

```
A=[0,1,2,3]
for A[2] in A:
    print(A[2], end="#")
```

**Output:**

0#1#1#3#

16: Write the output of the following code:

```
L = [None] * 10
print(len(L))
```

**Output:10**

## Programming Questions

Program 1: Write a program to calculate the sum of two numbers and print their sum.

```
num1 = 1.5
num2 = 6.3
sum = num1 + num2
print(sum)
```

**Output :**

7.8

Program 2: Write a program that accepts the radius of a circle and prints its area.

```
radius = float(input("Input the radius of the circle : "))
area = (22/7)*radius*radius
print (area)
```

**Output :**

Input the radius of the circle : 7.0

154.0

**Program 3:** Write a program to compute simple interest.

```
P = float(input("Enter Principal: "))
R = float(input("Enter Rate of Interest (in % per annum) : "))
T = float(input("Enter Time Period : "))

# Calculates simple interest
SI = (P * R * T) / 100
print(SI)
```

**Output :**

```
Enter Principal: 1000
Enter Rate of Interest (in % per annum) : 20
Enter Time Period : 3
600.0
```

**Program 4:** Write a program to take sides of a triangle as input and print its area.

```
a = float(input("Enter first side: "))
b = float(input("Enter second side: "))
c = float(input("Enter third side: "))

# calculate the semi-perimeter
s = (a+b+c)/2

# calculate the area
area = (s*(s-a)*(s-b)*(s-c))**0.5
print(area)
```

**Output :**

```
Enter first side: 5
Enter second side: 6
Enter third side: 7
14.696938456699069
```

**Program 5:** Write a program to find whether a given number is even or odd?

```
num = int(input("Enter a number: "))
if num % 2 == 0:
    print("Even Number")
```

```
else:  
    print("Odd Number")
```

**Output :**

Enter a number: 95  
Odd Number

**Program 6: Write a program to find the factorial of a number.**

```
num = int(input("Enter a number to find factorial: "))  
factorial = 1  
if num < 0:  
    print("Sorry, factorial does not exist for negative numbers")  
else:  
    for i in range(2,num+1):  
        factorial = factorial*i  
print("The factorial of is ",factorial)
```

**Output :**

Enter a number to find factorial: 6  
The factorial of is 720

**Program 7: Write a program to display the Fibonacci sequence up to n-th term where n is provided by the user**

```
first = 0  
second = 1  
counter = 0  
number_of_terms = int(input("Enter Number of terms : "))  
if number_of_terms <= 0:  
    print("Please enter a positive integer")  
else:  
    while counter < number_of_terms:  
        print(first, end = " ")  
        temp = first + second  
        first = second  
        second = temp  
        counter = counter + 1
```

**Output :**

Enter Number of terms : 10  
0 1 1 2 3 5 8 13 21 34

**Program 8:** Write a program in Python to check if a number entered by a user is a prime number or not.

```
number = int(input("Enter a positive number greater than one : "))
flag = 0
if number > 1:
    for i in range(2,number):
        if (number%i) == 0):
            flag = 1
            break
    if (flag==1):
        print("Not prime")
    else:
        print("Prime")
```

**Output :**

Enter a positive number greater than one : 79  
Prime

**Program 9:** Write a Program to check if the input year is a leap year or not

```
year = int(input("Enter a year : "))
if(((year % 4 == 0) and (year % 100 != 0)) or (year % 400 == 0)):
    print("Leap Year")
else:
    print("Non Leap Year")
```

**Output :**

Enter a year : 2100  
Non Leap Year

**Program 10:** Write a Python program to remove duplicates from a list.

```
L = [2, 4, 10, 20, 5, 2, 20, 4]
F = []
for n in L:
    if n not in F:
        F.append(n)
print(F)
```

**Output :**

[2, 4, 10, 20, 5]

**Program 11: Write a Python program to count the number of strings of length 2 or more and the first and last character are same from a given list of strings**

```
words = ['abc', 'xyz', 'aba', '1221']
count=0
for w in words:
    if len(w) > 1 and w[0] == w[-1]:
        count = count + 1
print(count)
```

**Output :**

2

**Program 12: Write a Python program to find the sum of all even numbers and all odd numbers from the given tuple.**

```
T = (8, 10, 13, 5, 6,9)
L= len(T)
even=0
odd=0
for i in range(L):
    if T[i]%2==0:
        even=even + T[i]
    else:
        odd = odd + T[i]

print("Sum of Even Numbers is = ", even)
print("Sum of Odd Numbers is = ", odd)
```

**Output:**

Sum of Even Numbers is = 24

Sum of Odd Numbers is = 27

**Program 13:** Write a program in Python, To input a list of numbers. Thereafter increments all even numbers by 1 and decrements all odd numbers by 1.

```
L = [8, 10, 13, 5, 6, 9]
print("Original List = ", L)
for i in range(len(L)):
    if L[i]%2==0:
        L[i]=L[i] + 1
    else:
        L[i] = L[i] - 1
print("Updated List = ", L)
```

**Output:**

Original List = [8, 10, 13, 5, 6, 9]

Updated List = [9, 11, 12, 4, 7, 8]

**Program 14:** Write a program in python, to input a dictionary Emp which contains eid and ename as key value pair and displays the name in uppercase of the values whose names are longer than 5 characters.

**For example:**

Emp = {101: 'Rahul', 102: 'Hardik', 103: 'Virat', 104:'Jaspreet'}

**The output should be:**

**Hardik**

**Jaspreet**

```
Emp = {}
c='y'
while c in 'yY':
    eid = int(input("Enter the employee id: "))
    ename = input("Enter the name of employee: ")
    Emp[eid] = ename
    c = input("Do you want to add more elements(y/n): ")
print("Dictionary is : ", Emp)
print("Employee name longer than 5 characters: ")
for name in Emp.values():
```



```
if len(name)>5:  
    print(name.upper())
```

**Output:**

Enter the employee id: 101  
Enter the name of employee: Rahul  
Do you want to add more elements(y/n): y  
Enter the employee id: 102  
Enter the name of employee: Hardik  
Do you want to add more elements(y/n): y  
Enter the employee id: 103  
Enter the name of employee: Virat  
Do you want to add more elements(y/n): y  
Enter the employee id: 104  
Enter the name of employee: Jaspreet  
Do you want to add more elements(y/n): n  
Dictionary is : {101: 'Rahul', 102: 'Hardik', 103: 'Virat', 104: 'Jaspreet'}  
Employee name longer than 5 characters:  
HARDIK  
JASPREET

**Question : Write the output of the following code:**

```
a = "Year 2022 at All the best"  
a = a.split('2')  
b = a[0] + ". " + a[1] + ". " + a[3]  
print (b)
```

**Output :**

Year . 0. at All the best

**Question :**

**Match the following: -**

1) Returns True if all characters in the string are digits	(a). count()
2) Returns the number of times a specified value occurs in a string	(b). extend()
3) Splits the string at the specified separator, and returns a list	(c). append()
4) Add the elements of a list (or any iterable), to the end of the current list	(d). isdigit()
5) Removes the first item with the specified value	(e). split()
	(f). pop()
	(g). remove()

**Answer:**

**1 - D, 2 - A, 3 - E, 4 - B, 5 - G**

**Question: Write the output of the following code:**

```

tuple1 = (11, 22, 33, 44, 55 ,66)
list1 =list(tuple1)
new_list = []
for i in list1:
    if i%2==0:
        new_list.append(i)
new_tuple = tuple(new_list)
print(new_tuple)

```

**Output :**

(22, 44, 66)

**Question: Write the output of the following code:**

```
N={'A':'B','C':'D'}
for k,v in N.items():
    print(k,v)
```

**Output :**

A B

C D

### Assertion and Reason Type Questions

**1. Assertion (A): Strings in Python are immutable.**

**Reason (R): Once a string object is created, it cannot be changed.**

- A. Both A and R are true and R is the correct explanation for A
- B. Both A and R are true and R is not the correct explanation for A
- C. A is True but R is False
- D. A is False but R is True

**Answer : A. Both A and R are true and R is the correct explanation for A**

**2. Consider the code given below:**

```
x=5
print(x==5)
```

**A: Output of above python code is True.**

**R: '==' is logical Operator.**

- A. Only A is True.
- B. Only R is True.
- C. Both A and R are True, but R is not the correct reason for A.
- D. Both A and R are True, but R is the correct reasoning for A.

**Answer : A. Only A is true.**

**3. Assertion (A): The in operator in python is membership Operator.**

**Reason (R): in operator in Python is used searching for specific elements within sequences like lists and strings etc.**

- A. Both A and R are true and R is the correct explanation for A
- B. Both A and R are true and R is not the correct explanation for A
- C. A is True but R is False
- D. A is False but R is True

**Answer : A. Both A and R are true and R is the correct explanation for A**

**4. Assertion (A): Modules in Python are reusable pieces of code that can be imported into other Python scripts.**

**Reason (R): import statement must be the first line of the program**

- A. Both A and R are true and R is the correct explanation for A
- B. Both A and R are true and R is not the correct explanation for A
- C. A is True but R is False
- D. A is False but R is True

**Answer: A is True but R is False**

**5. Assertion (A): pop() used to delete the last element from the dictionary.**

**Reason (R): popitem() used to delete the specific element from the dictionary.**

- A. Both A and R are true and R is the correct explanation for A
- B. Both A and R are true and R is not the correct explanation for A
- C. A is True but R is False
- D. Both A and R are False

**Answer: Both A and R are False**

**6. Assertion (A): extend() method is used to merge two dictionaries.**

**Reason (R): items() method is used to display the keys and also the values of the dictionary.**

- A. Both A and R are true and R is the correct explanation for A
- B. Both A and R are true and R is not the correct explanation for A
- C. A is True but R is False
- D. A is False but R is True

**Answer: A is False but R is True**

**7. Assertion (A): L.len() statement used to return the total number of elements in the list.**

**Reason (R): count() method is used to find the total number of elements in the list.**

- A. Both A and R are true and R is the correct explanation for A
- B. Both A and R are true and R is not the correct explanation for A
- C. A is True but R is False
- D. Both A and R are False

**Answer: Both A and R are False**

---

## UNIT 3 : Society, Law and Ethics

### Digital Footprints

- Digital footprints are a set of footprints (trackable information or activity) left behind while using any digital device such as smartphone, desktop or laptop computers, or performing activities such as browsing the internet, posting on social media, playing a game, editing a file etc.



- In other words, it can be considered as the data trail – intentional and unintentional - that is left behind while surfing the web.
- Digital footprints are the information that others can see or collect about you.

Event	Footprints(information)
Visiting a website	IP address, cookies, browsing history, interests
Social media post	Location, Username, personal Photos
Sending email	IP address, Email address
Online shopping	IP address, User behavior, Payment information

### How are digital footprints created ?

Digital footprint are of two types

#### 1. Active Digital Footprint

An active digital footprint is intentionally/deliberately shared by the user, either by using social media sites, emails or by using websites.

When is an active digital footprint created?

- Posting on social media: Sharing updates, photos, videos, and comments on platforms like Facebook, Instagram, Twitter, etc.
- e-commerce: Providing personal and payment information when shopping online.

#### DIGITAL FOOTPRINT

How to Preserve Your Digital Footprint?



## 2. Passive Digital Footprint

Information that is unintentionally shared by the user creates a Passive Digital Footprint.

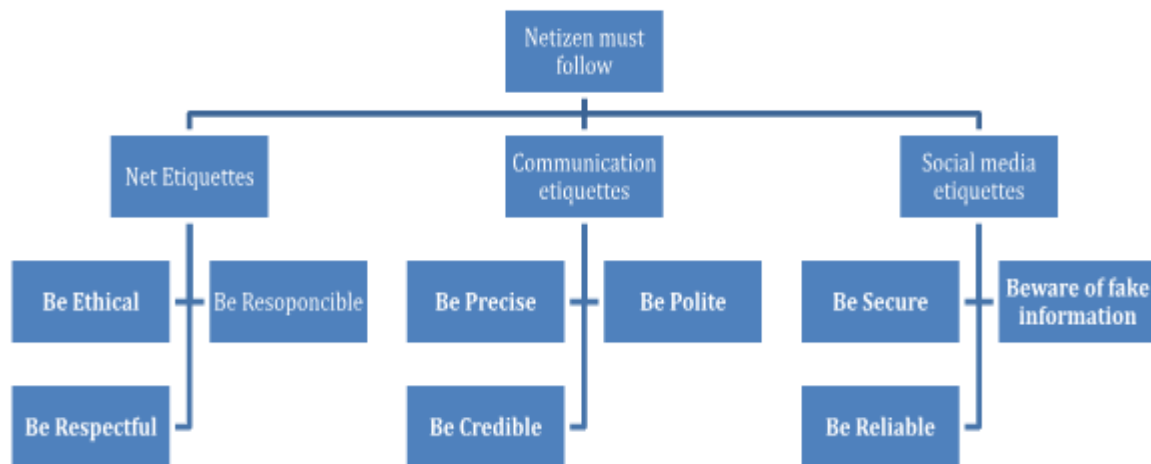
**When are Passive digital footprint created?**

- **Cookies and browsing history:**  
Websites use cookies to track browsing behavior, preferences, and interactions on their site.
- **IP address tracking:**  
Websites can log your IP address, which can reveal approximate location and the internet service provider.

**How to minimize passive digital footprint?**

We can adjust privacy settings, clear cookies and browsing history regularly, and use tools that block online tracking.

## Digital Society and Netizen



Anyone who uses digital technology along with the Internet is a digital citizen or netizen.

A responsible netizen must follow

1. Net etiquettes
2. Communication etiquettes
3. Social media etiquettes

## Net etiquettes

*Etiquettes* : set of rules that govern appropriate and respectful conduct.

Netiquette, (Internet etiquette) refers to the set of guidelines and rules for polite, respectful, and responsible behavior while using digital communication platforms for communicating online. We should follow certain etiquettes during our social interactions.

- **Be Ethical**
  - *ethical : morally correct*
  - No copyright violation: we should not use copyrighted materials without the permission of the creator or owner.
  - Share the expertise: it is good to share information and knowledge on the Internet so that others can access it.
- **Be Respectful**
  - Respect privacy
    - In the physical world : Privacy is the state of being alone, or freedom from disturbance/intrusion.
    - In the Digital world also everyone has the right to privacy and the freedom of personal expression.
  - Respect diversity: In a group or public forum, we should respect the diversity of the people in terms of knowledge, experience, culture and other aspects.
- **Be Responsible**
  - Avoid cyber bullying
    - *bully : to use your strength or power to hurt or frighten somebody who is weaker*
  - In Cyber world
    - Cyberbullying is bullying (to harass, threaten, embarrass) with the use of digital devices like cell phones, computers, and tablets.
- **Don't get involved in trolling.**

An internet troll is a person who posts inflammatory or off topic messages in an online community, just for amusement or seeking attention. The best way to discourage trolls is not to pay any attention to their comments

## Digital Communication Etiquettes (Rules for good Digital Communication)

- **In Physical world**
  - communication : the act of sharing or exchanging information, ideas or feelings
- **In Digital world**
  - Digital Communication : Digital communication includes email, texting, instant messaging, talking on the cell phone, audio or video conferencing, posting on forums, social networking sites, etc.
- **Be Precise**
  - We should be clear and accurate while communicating online. We should compress very large attachments before sending.
- **Be Polite**
  - Polite : showing respect for others
  - We should be polite and non-aggressive in our communication



- **Be Credible**

- Credible : that can be believed (विश्वसनीय)
- We should be cautious while making a comment, replying or writing an email or forum post as such acts decide our credibility over a period of time.

## Social Media Etiquettes

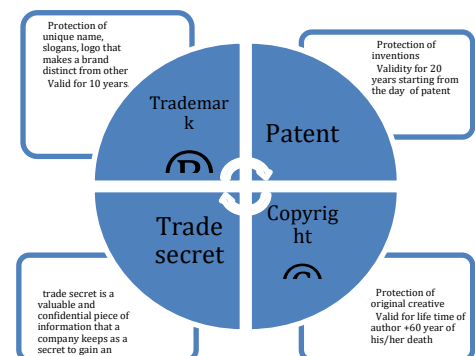
- Social media is a collective term for websites and applications that facilitates the sharing of ideas, thoughts, and information  
Example : Facebook, Twitter, Instagram
- **Be Secure**
  - Choose password wisely
  - Know who you befriend
  - Choose Friends Wisely
- **Beware of fake information**
  - we should be able to figure out whether a news, message or post is genuine or fake by checking Facts (PIB Fact Check)
- **Be Reliable**
  - Think before uploading

## Data Protection

- Data or information protection means safeguarding and preserving the privacy of data stored digitally.
- Data that cause substantial harm, embarrassment, inconvenience and unfairness to an individual, if breached or compromised is called **sensitive information**.  
For example, financial information, personal information etc.
- The main goal of data protection is to ensure that individuals' personal information is processed and handled in a secure and lawful manner, protecting their rights and privacy..

## Intellectual Property Right (IPR)

- Intellectual property (IP) refers to the ownership of an idea or design by the person who came up with it.
- Intellectual property (IP) refers to innovation such as inventions; literary and artistic works; designs; and symbols, names and logos.
- Intellectual Property is legally protected through copyrights, patents, trademarks, etc



## Copyrights

- Copyright grants legal rights to creators for literary, dramatic, musical, artistic works, photograph, audio recordings, video recording, computer software's.
- The rights include right to copy (reproduce) a work, right to distribute copies of the work to the public, and right to publicly display or perform the work.



Patent



Copyright



Trademark

## Patent

- A patent is usually granted for inventions. When a patent is granted, the owner gets an exclusive right to prevent others from using, selling, or distributing the protected invention.
- Patent gives full control to the patentee to decide whether or how the invention can be used by others.
- Example : Pen with scanner (with a machine as small as a pen, we can transfer text from paper directly into a computer)

- granted for inventions
- protect inventions and innovations
- Patent owners have the exclusive rights to make, use, and sell the patented invention
- Validity: 20 years from the date of filing patent

- Granted for original creative works
- protects original works of authorship
- Owner has exclusive rights to reproduce, distribute, perform, display
- Validity: author's lifetime plus a certain number of years after their death

- granted for distinguishing goods or services in the marketplace
- protect brand names, logos, symbols, phrases
- Owner can preventing others from using a confusingly similar mark in the same or related field.
- Validity: indefinitely as long as they are in use

## Trademark

1. Trademark is a visual symbol, name, design, slogan, label, etc., that distinguishes the brand or commercial enterprise from other brands or commercial enterprises.
2. Example trademark of Gmail, Macdonald etc



## Violation of IPR

When we use some other intellectual property (idea,image,logo,trademark) without taking consent or permission from the owner. IPR violation may occur in following ways:

## Plagiarism

Presenting someone else's idea or work as one's own idea or work is called plagiarism. If we copy some contents from the Internet, but do not mention the source or the original creator, then it is considered as an act of plagiarism.

## Copyright Infringement

- Infringement = उल्लंघन
- Copyright infringement is when we use another person's work without obtaining their permission to use or we have not paid for it, if it is being sold.

## Trademark Infringement

- Trademark Infringement means unauthorized use of another's trademark on products and services.

## Free and Open Source Software (FOSS) and Licensing :

### Free and Open Source Software (FOSS)

- Free and Open Source Software (FOSS) is a type of software which is free and the source code is publicly available so that anyone can use it, study it, and even change or improve it.
- The goal is to encourage collaboration among users and developers to make the software better together.
- Examples of FOSS include Ubuntu operating system, Python programming language, Libreoffice, Openoffice, and Mozilla Firefox web browser.

### Freeware

- Sometimes, software is freely available for use but source code may not be available. Such software is called freeware. Examples of freeware are Skype, Adobe Reader, etc.

### Proprietary

- When the software to be used has to be purchased from the vendor who has the copyright of the software, then it is proprietary software.
- The source code is not publicly available. Only the company which has developed it, can modify it.
- These software's are developed and tested by individuals or the organization by which it is owned, not by the public.
- Examples of proprietary software include Microsoft Windows, Quickheal, etc.

## License

*License : An official document that gives you permission to own, do, or use something.*

- A public license or public copyright licenses is a license by which a copyright holder as licensor can grant additional permissions to others to use and even modify the content.
- The GNU General public license (GPL) and the Creative Commons (CC) are two popular categories of public licenses.

### **Creative Commons (CC)**

- CC is used for all kinds of creative works like websites, music, film, literature, etc. CC enables the free distribution of an otherwise copyrighted work. It is used when an author wants to give people the right to share, use and build upon a work that they have created.

### **GNU General public license (GPL)**

- The GNU General Public License (GNU GPL or simply GPL) is primarily designed for providing a public license to a software. It guarantees end users the freedom to run, study, share, and modify the software.

### **Apache**

- The Apache License is a type of free software license created by the Apache Software Foundation.
- It is very flexible and permissive, which means people can use the software in any way they want, share it with others, and even make changes to it.
- No need to worry about paying any fees or royalties for using or sharing the software. It gives a lot of freedom to users and encourages collaboration and sharing within the software community.

### **Cyber Crime:**

- Any criminal, illegal or harmful activity conducted using computers, the internet, or other digital device is referred to as Cyber Crime.
- These activities are committed by individuals or groups to steal or harm someone else's data, privacy, or online safety.
- Here are some examples of cybercrimes: Cyber bullying, online scams, hacking, stalking, ransomware attack, phishing etc.

### **Hacking:**

- Hacking is unauthorized access to a computer or a network with the intention of committing a crime.
- It can also be explained as the act of accessing computer systems, networks, or digital devices in a skilful and creative manner to explore and find some security loopholes in order to gain access to confidential information.
- The process of gaining unauthorized access to a computer system, group of systems or an organization's data is known as hacking.
- The person engaged in these activities is generally known as a Hacker.

### **Eavesdropping:**

- Eavesdropping is to intercept and listen to private electronic communications, such as emails, instant messages, or phone calls, without the consent of the parties involved.
- It can be done via hacking or surveillance techniques to access the conversations. The main purpose of eavesdropping is to steal data.

## Phishing:

- Phishing is a cybercrime where criminals attempt to deceive Individuals into revealing sensitive information like passwords or credit card details by posing as trustworthy entities through fake emails, websites, or messages.
- For example an email of winning a lottery and asking you to fill your bank details.

## Fraud Emails:

- Fraudulent emails are cybercrimes where bad people send fake mails that try to trick people to get personal information, passwords, or money.
- These dishonest emails may pretend to be from a popular website, but it's essential to be cautious and never share sensitive details with unknown senders.

## Ransomware:

- Ransomware is a cybercrime where bad people create harmful software that locks or encrypts important files on a computer.
- They then demand ransom from the computer's owner to unlock the files and make them accessible again.
- It's essential to be careful while using computers and not click on suspicious links or download unknown files to avoid ransomware attacks.



## Cyber Trolls:

Trolls are visitors who leave inflammatory comments in public comment sections. Whether they comment on blog posts or online news sites, they are looking to grab the attention of other visitors and disrupt discussion that would otherwise be about the page's content.

## Cyber Bullying:

- Cyber bullying is bullying with the use of digital technologies. It is to intimidate, harass, demean, defame or humiliate others repeatedly using digital platforms such as the internet, social media, phone, internet, instant messengers etc.
- It can cause emotional distress, anxiety, and damage self-esteem. Examples include: posting embarrassing photos of someone on social media, sending hurtful messages.



## Cyber Safety:

- Cyber Safety refers to the practice of protecting oneself, one's information, and digital assets from potential internet threats or online threats. Cyber Security is to protect users from harmful online activities.
- The aim of cyber safety is to promote responsible and secure online behavior to ensure a safe experience for everyone.

## Safely Browsing the Web:

These days working on the web or the internet have become very common and inevitable. We must be aware of the threats while browsing the web.

- To safely browsing on the web we should know the following things:
  1. What can be the possible dangers/threats?
  2. How can we avoid these?

## Identity Protection while using the internet:

We browse the internet these days for a variety of reasons via providing our personal information to sell or purchase goods on the internet, on social media platforms and so on.

- This information can be used in a fraudulent way. Fraud which involves another's identity to steal money or to gain other benefits is known as **Identity Theft or Identity Fraud**.
- It can be of Financial theft , criminal theft/ medical theft.

## Confidentiality of Information:

- Confidentiality of information refers to the protection and safeguarding of sensitive or private data from unauthorized access, disclosure, or use.
- The owner of the information or the data has to decide who can have the access or use the data and who can't.
- To ensure confidentiality, organizations and individuals can implement various security measures, including: Access controls, Encryption, Physical security, Regular security audits.

## Malware

- Malware (malicious software) is any software/program that is designed to damage and destroy computers and computer systems.
- Computer Malware is like bad software that can cause problems for your computer or device. It comes in different forms, like viruses, trojans, and adware.

## Virus

- A computer virus is a malware (malicious computer code) that spreads from one device to another. They are like digital germs that infect and harm your computer by spreading from one file to another. After entering a computer, a virus attaches itself to another program (like a document) in such a way that execution of the host program triggers the action of the virus simultaneously.
- It can self-replicate, inserting itself onto other programs or files, infecting them in the process. Most viruses perform actions that are malicious in nature, such as damaging programs, deleting or destroying data.
- Viruses spread when the software or documents they get attached to are transferred from one computer to another using a network, file sharing methods, or through e-mail attachments.

## **Trojan horse :**

- It is a file or program, or piece of code that appears to be legitimate and safe, but is actually malware.
- Trojan horse malware is generally designed to spy on victims or steal data. These programs perform some malicious activities like upload (send) some security files and information from the computer and at the same time download some unwanted files onto the computer.

## **Adware**

- Adware (or advertising software) is the term used for various pop-up advertisements that show up on your computer or mobile device.
- Adware has the potential to become malicious and harm your device by slowing it down, hijacking your browser and installing viruses

## **E-waste management: proper disposal of used electronic gadgets.**

### **E-waste**

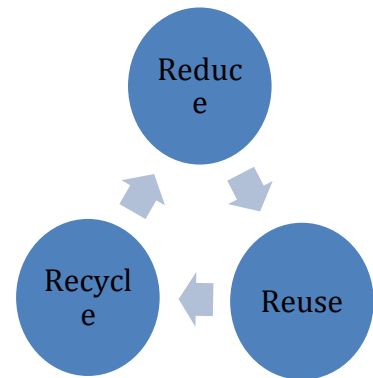
- E-waste or Electronic waste includes electric or electronic gadgets and devices that are no longer in use. Hence, discarded computers, laptops, mobile phones, televisions, tablets, music systems, speakers, printers, scanners etc. constitute e-waste when they are near or end of their useful life.
- Globally, e-waste constitutes more than 5 percent of the municipal solid waste. Therefore, it is very important that e-waste is disposed of in such a manner that it causes minimum damage to the environment and society.
- When e-waste is carelessly thrown or dumped in dumping grounds, certain elements or metals used in production of electronic products cause air, water and soil pollution. This is because when these products come in contact with air and moisture, they tend to leach. As a result, the harmful chemicals seep into the soil, causing soil pollution. Further, when these chemicals reach and contaminate the natural ground water, it causes water pollution as the water becomes unfit for humans, animals and even for agricultural use. When dust particles loaded with heavy metals enter the atmosphere, it causes air pollution as well.

Some of the feasible methods of e-waste management are reduce, reuse and recycle.

## E-waste management cycle:

### Reduce:

- We should try to reduce the generation of e-waste by purchasing the electronic or electrical devices only according to our need.
- Also, they should be used to their maximum capacity and discarded only after their useful life has ended. Good maintenance of electronics devices also increases the life of the devices.



### Reuse:

- It is the process of re-using the electronic or electric waste after slight modification.
- The electronic equipment that is still functioning should be donated or sold to someone who is still willing to use it.
- The process of re-selling old electronic goods at lower prices is called refurbishing.

### Recycle:

- Recycling is the process of conversion of electronic devices into something that can be used again and again in some or the other manner.
- Only those products should be recycled that cannot be repaired, refurbished or re-used.
- To promote recycling of e-waste many companies and NGOs are providing door-to-door pick up facilities for collecting the e-waste from homes and offices.

## Information Technology Act

- The Information Technology Act (IT Act) is an Indian law that was enacted in the year 2000 to provide legal recognition to electronic transactions and to address issues related to electronic commerce, data protection, and cybercrimes.
- The IT Act aims to facilitate electronic communication and transactions while ensuring the security and confidentiality of electronic information.  
Example: Suppose you want to buy a smartphone online from an e-commerce website. The transaction involves entering your personal and financial details on the website, such as your name, address, credit card number, and CVV (Card Verification Value).

### Key Points

1. **Legal Recognition of Electronic Transactions:** The IT Act gives legal validity to electronic records, including online transactions. So, when you make a purchase online and receive an electronic receipt, it is legally recognized and can be used as evidence in case of any disputes.
2. **Electronic Signatures:** The IT Act recognizes electronic signatures as equivalent to physical signatures, making contracts and agreements signed electronically



legally binding. When you electronically sign the purchase agreement on the website, it holds the same legal weight as a physical signature.

3. **Data Protection and Privacy:** The IT Act includes provisions for data protection and privacy. The e-commerce website is obligated to take necessary measures to protect your personal and financial information from unauthorized access or misuse. They must have a privacy policy in place, and any data collection and processing must be done with your consent.
4. **Cybercrime Provisions:** The IT Act addresses cyber crimes such as hacking, unauthorized access, and data theft. If someone tries to steal your credit card information during the online transaction, the IT Act provides a legal framework to prosecute the offender.

## Technology and society: Gender and disability issues while teaching and using computers

### Gender Issues

- Preconceived notions
  - Notions like boys are better at technical things, girls are good at humanities etc
- Interest development from primitive years
  - During primitive years children often played games on the computers and smartphones. Most of the games are boys centric that increase the interest of boys in computers.

### Disability Issues

- Unavailability of teaching material / aids

## True or False type questions

**Q. Digital footprints are only created by social media activity, such as posting photos and comments.**

**Answer:** False.

**Q. If you delete your information from the internet then your digital footprint is also deleted**

.

**Answer:** False. Deleted information is difficult to trace but it may be accessible through backups or cached versions.

**Q. Plagiarism is considered a violation of intellectual property rights, even if the copied material is not used for commercial purposes.**

**Answer:** True.

**Q. The Creative Commons license allows creators to retain their copyright while permitting others to use their work under certain conditions.**

**Answer:** True.

**Q. Viruses can spread through email attachments and infect computers when the attachment is opened.**

**Answer:** True.

**Q. Trojans are self-replicating programs that can spread across a network without user intervention.**

**Answer:** False.

**Q. Phishing is a form of cybercrime where attackers attempt to deceive individuals into disclosing sensitive information through fraudulent emails or websites.**

**Answer:** True.

**Q. Ransomware is a type of malicious software that locks users out of their systems or encrypts their files until a ransom is paid.**

**Answer:** True.

### **Multiple Choice questions:**

**Q. What is a digital footprint?**

- A. The mark left on the floor by a digital device
- B. A trail of personal information and online activities
- C. A type of digital currency used for online transactions
- D. A digital signature used for secure authentication

**Answer: B**

**Q. Why is it essential to manage your digital footprint?**

- A. To avoid being tracked by internet service providers
- B. To increase internet speeds and reduce latency
- C. To prevent cyber-attacks and data breaches
- D. To conserve digital resources and reduce energy consumption

**Answer: C**

**Q. Which of the following is an example of a passive digital footprint?**

- A. Posting a status update on social media
- B. Sending an email to a friend
- C. Deleting browser history and cookies
- D. Websites tracking your online activities using cookies

**Answer: D**

**Q. Which of the following is an example of an active digital footprint?**

- A. Websites collecting data about your browsing habits
- B. An online forum post that includes personal information
- C. Tracking cookies left on your computer by websites
- D. Using private browsing mode to browse the internet

**Answer: B**

**Q. What does "net etiquette" refer to?**

- A. The rules and conventions for using the internet respectfully and responsibly.
- B. The study of networking technologies and protocols.
- C. A type of online gaming community.
- D. A web design technique for creating interactive websites.

**Answer: A**

**Q. What is the appropriate action to take when you receive a suspicious link or message?**

- A. Click on the link to see where it leads.
- B. Report it to the platform or email provider as spam.
- C. Share the link with friends to get their opinion.
- D. Respond to the message and inquire about its source.

**Answer: B**

**Q. What should you do if you come across offensive content or cyberbullying online?**

- A. Engage in the conversation to confront the offenders.
- B. Ignore it and move on to avoid getting involved.
- C. Report the content to the website or platform administrators.
- D. Retaliate with offensive content to defend yourself or others.

**Answer: C**

**Q. How should you handle disagreements in digital communication?**

- A. Respond with offensive remarks to prove your point.
- B. Engage in a public argument to gain more attention.
- C. Stay calm and express your viewpoint respectfully.
- D. Share the disagreement with others to create awareness.

**Answer: C**

**Q. Which of the following refers to the legal protection of original creations, such as artistic works and literary pieces?**

- A. Data Encryption

- B. Intellectual Property Rights (IPR)
- C. Data Privacy
- D. Data Breach

**Answer: B**

**Q. What type of intellectual property right grants exclusive rights to inventors for their inventions?**

- A. Copyright
- B. Trademark
- C. Patent
- D. Creative Commons

**Answer: C**

**Q. Open-source software is distributed with a license that allows users to:**

- A. Use the software without any restrictions.
- B. Modify and distribute the software freely.
- C. Sell the software for a profit without attribution.
- D. Use the software for personal use only.

**Answer: B**

**Q. Which form of intellectual property right protects logos, brand names, and distinctive signs used in commerce?**

- A. Copyright
- B. Patent
- C. Trade Secret
- D. Trademark

**Answer: D**

**Q. Plagiarism is the act of:**

- A. Copying and using someone else's work without proper attribution.
- B. Using open-source software without a valid license.
- C. Distributing copyrighted material for free online.
- D. Creating derivative works based on public domain content.

**Answer: A**

**Q. What is malware?**

- A. Software designed to protect computer systems from threats.

- B. A type of software that enhances computer performance.
- C. Malicious software designed to harm computer systems and data.
- D. Software that helps in data recovery after a system crash.

**Answer: C**

**Q. Which type of malware disguises itself as legitimate software to trick users into installing it?**

- A. Virus
- B. Trojan
- C. Adware
- D. Spyware

**Answer: B**

**Q. What does hacking involve?**

- A. Legally accessing computer systems
- B. Unauthorized access to computer systems
- C. Protecting computer networks
- D. Improving software efficiency

**Answer: B**

**Q. Eavesdropping is the act of:**

- A. Intercepting and listening to private communications
- B. Sending fraudulent emails
- C. Installing firewalls on computers
- D. Monitoring social media posts

**Answer: A**

**Q. Ransomware is malicious software that:**

- A. Protects computer systems from cyber trolls
- B. Encrypts files and demands payment for decryption
- C. Monitors private communications
- D. Enhances internet browsing speed

**Answer: B**

**Q. Cyber bullying involves:**

- A. Legitimate criticism on the internet
- B. Sending fraudulent emails to individuals
- C. Harassing, intimidating, or threatening others online
- D. Collaborative online projects

**Answer: C**

**Q. What's the best way to protect against cybercrime?**

- A. Sharing sensitive information online
- B. Using weak passwords
- C. Regularly updating security software
- D. Ignoring software updates

**Answer: C**

### Very short answer type

**Q. What is a digital footprint?**

**Answer:** A digital footprint is the trail of data and information left by an individual's online activities.

**Q. How is a digital footprint created?**

**Answer:** Digital footprints are created through online interactions, such as social media posts, website visits, and online purchases.

**Q. What are Intellectual Property Rights (IPR)?**

**Answer:** IPR refers to legal rights that protect creations of the mind, such as inventions, artistic works, and brand names.

**Q. What is the purpose of Creative Commons licenses?**

**Answer:** Creative Commons licenses provide creators with a way to share their work under specific terms and conditions.

**Q. What is malware?**

**Answer:** Malware is malicious software designed to harm computer systems or steal data.

**Q. How does a Trojan operate?**

**Answer:** A Trojan disguises itself as legitimate software to trick users into installing it and then performs malicious actions.

**Q. How can users protect themselves from malware?**

**Answer:** Users can install reputable antivirus software, keep systems and applications up to date, and avoid downloading files from unknown sources.

**Q. What is the Information Technology Act (IT Act)?**

**Answer:** The Information Technology Act is a legal framework that addresses electronic transactions, cybersecurity, and digital data protection.

**Q. What are some objectives of the IT Act?**

**Answer:** The objectives of the IT Act include providing legal recognition for electronic transactions, facilitating e-governance, and preventing cyber crimes.

**Q. How can technology help address gender disparities in education?**

**Answer:** Technology can provide equal learning opportunities for all genders and promote access to education and professional opportunities.

### Short answer type question

**Q. How does an individual's digital footprint impact their online privacy and personal brand? Provide an example to support your answer.**

**Answer:** An individual's digital footprint can expose personal information and influence how they are perceived online. For instance, sharing inappropriate content on social media may harm their reputation and affect future opportunities.

**Q. Explain how digital footprints are created and expanded through online interactions.**

**Answer:** Digital footprints are formed by an individual's online activities, such as social media posts, website visits, online purchases, and interactions with others. Each online engagement adds to the footprint, shaping their online identity.

**Q. How can individuals and organizations ensure they are not violating intellectual property rights while using copyrighted materials for educational or commercial purposes?**

**Answer:** Individuals and organizations should seek proper licensing, obtain permission from the copyright owner, or use materials that are explicitly labeled for reuse under Creative Commons licenses.

**Q. Discuss the importance of open-source software in fostering innovation and collaboration within the technology community.**

**Answer:** Open-source software encourages sharing and collaboration, leading to faster development, higher quality, and a more extensive range of software solutions accessible to everyone.

**Q. How can users differentiate between a legitimate software application and a potentially harmful trojan or malware?**

**Answer:** Users should only download software from reputable sources, check reviews, and verify the publisher's authenticity to avoid installing trojans or malware.

## Case Study based question:

### Case Study1 :

Shreya is a high school student who is an active user of social media platforms. She frequently posts photos, videos, and personal information online. One day, she notices that some of her private photos have been shared without her permission. Sarah becomes concerned about her digital footprint and its potential consequences.

**Q1. What is a digital footprint, and how is Shreya's online activity contributing to it?**

**Q2. Identify the potential risks and consequences Shreya may face due to her unmanaged digital footprint.**

**Q3. As a friend of Shreya, what advice would you give her to better manage and protect her digital footprint to prevent future incidents?**

### Answers

**A1.** A digital footprint is the collection of data left behind by an individual's online activities. Shreya's frequent social media posts, photos, and videos contribute to her digital footprint.

**A2.** Shreya may face privacy breaches, cyberbullying, or negative reputational impact due to her unmanaged digital footprint.

**A3.** Shreya should review her privacy settings, limit sharing of personal information, and regularly audit her online content to manage her digital footprint effectively

### Case Study 2:

A software developer named Ravi created a new mobile application and shared it with colleagues and friends. However, Ravi later discovered that someone had copied the entire code of the application and released it as their own. Ravi is unsure of what legal actions to take to protect their intellectual property rights.

**Q1. Describe the different types of intellectual property rights, including copyrights, patents, and trademarks, and their significance for Ravi's mobile application.**

**Q2. Explain the concept of plagiarism and copyright infringement in the context of the copied mobile application.**

**Q3. What are the possible legal measures Ravi can take to address the violation of their intellectual property rights and seek appropriate recourse?**

### Answers:



**A1.** Copyrights protect the code and creative aspects of Ravi's mobile application, while patents safeguard its novel functionality. Trademarks protect the app's branding.

**A2.** Plagiarism occurs when someone copies another's work without permission, and copyright infringement happens when copyrighted material is used without authorization.

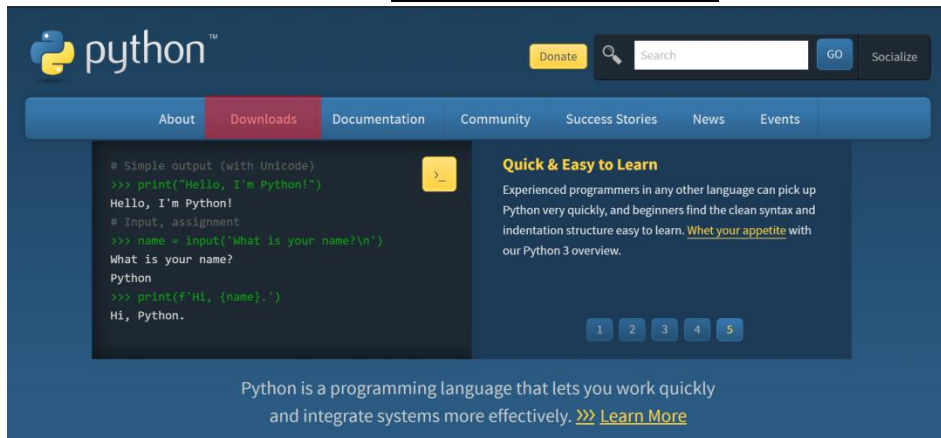
**A3.** Ravi can take legal action against the infringer to protect the intellectual property rights.

# Appendix-1

## Installing Python

1. Visit the official website of Python

<https://www.python.org>

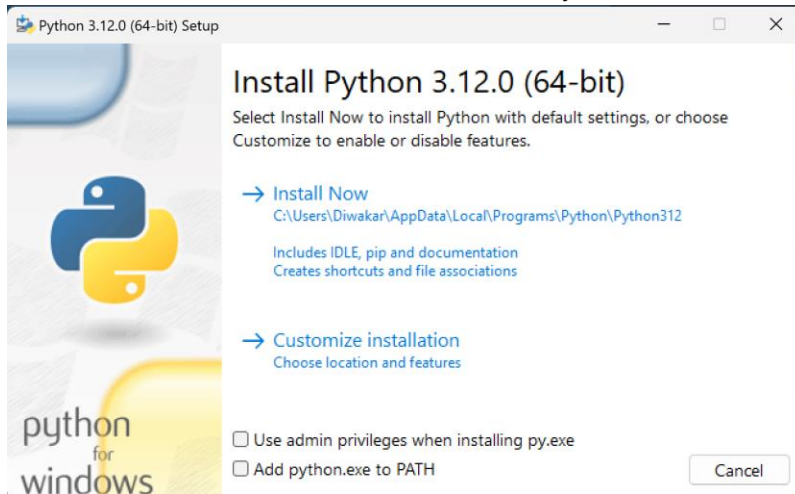


2. Go to the **Downloads** section and choose the Python version for your Operating System (Windows/macOS etc)



3. Click on Python 3.xx.x (here it is Python 3.12.0) where x may be any sub-version of Python.
4. Your download will start automatically, save the file at a location of your choice.

5. Double click on the saved file to start the Python Installation.



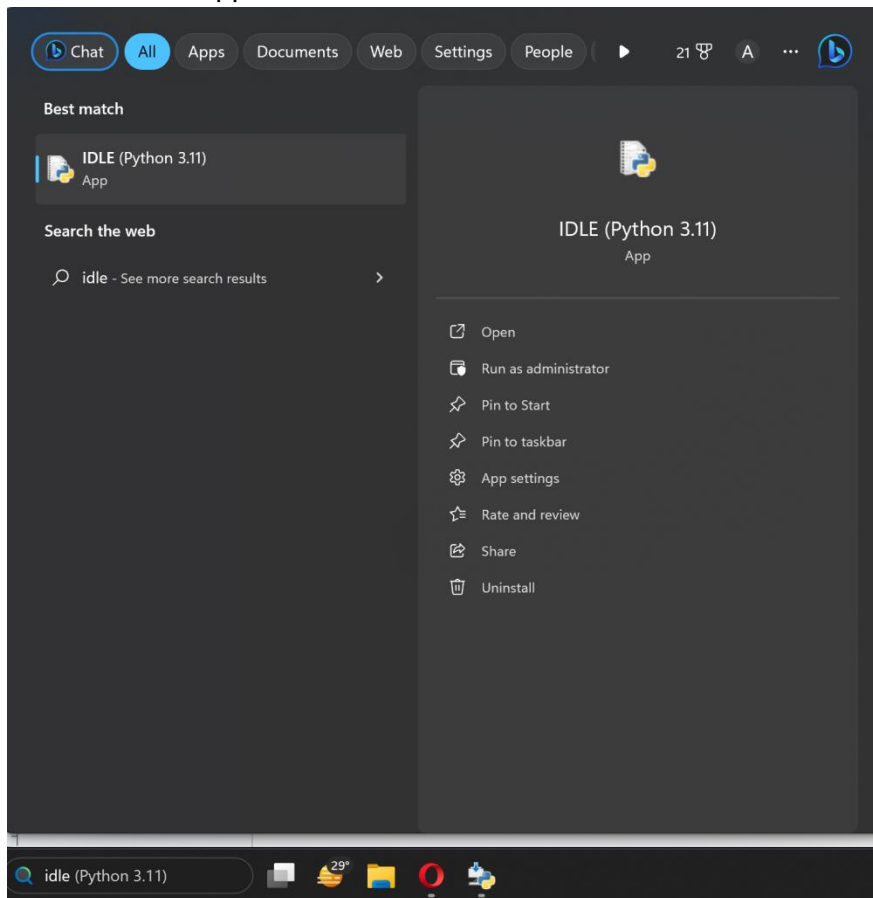
6. Click on '**Install Now**' to start the installation of Python in the default location. Alternatively, the location may be changed by clicking on the 'Customize Installation' option.
7. **Wait** for the installation to finish.

# Appendix - 2

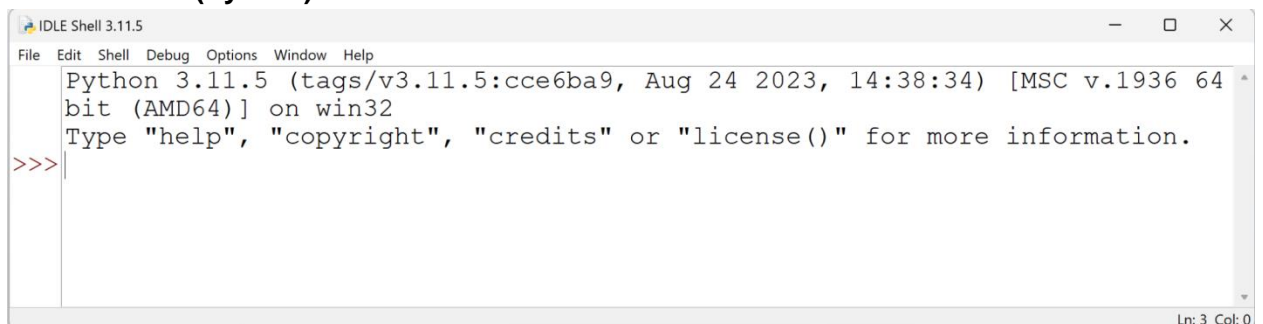
## Running Python

### Using IDLE in Interactive Mode

1. IDLE may be started by typing '**IDLE**' on the Windows Search Bar, or choosing IDLE from the list of applications in the Windows.

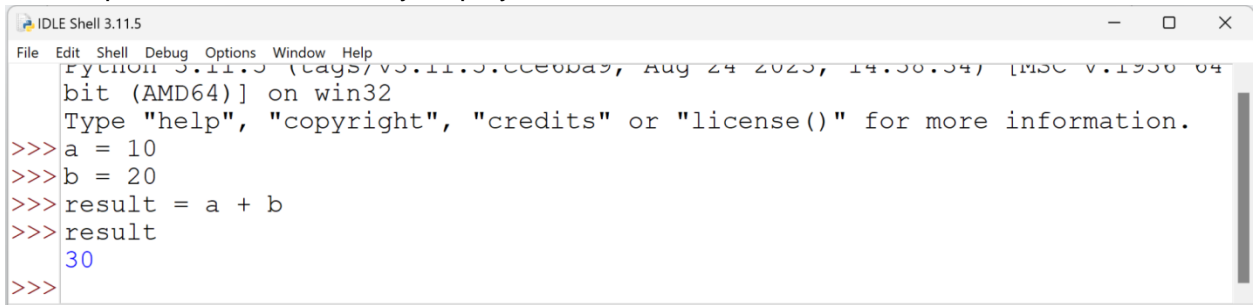


2. Click on **IDLE (Python)** to start IDLE.



3. >>> shows the **prompt** where Python statements can be written in the Interactive Mode.

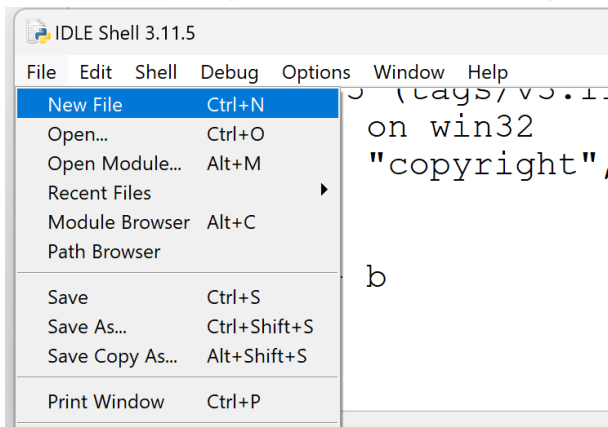
- The outputs will be immediately displayed after the entered statement.



```
Python 3.11.5 (tags/v3.11.5:ccc0b99, Aug 24 2023, 14:38:34) [MSC v.1936 64-bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> a = 10
>>> b = 20
>>> result = a + b
>>> result
30
>>>
```

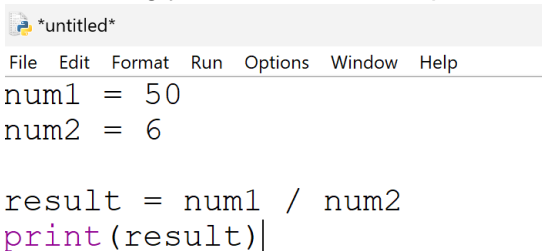
## Using Script Mode in IDLE

- To start writing code in the Script Mode, start IDLE in Interactive Mode.
- On the Menu Bar, choose **File > New File**, to open a new window.



```
Python 3.11.5 (tags/v3.11.5:ccc0b99, Aug 24 2023, 14:38:34) [MSC v.1936 64-bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> a = 10
>>> b = 20
>>> result = a + b
>>> result
30
>>>
```

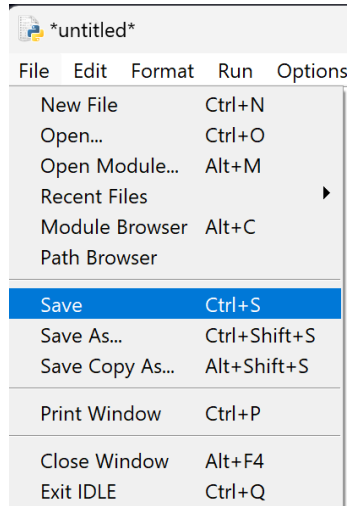
- Start writing your code in the script window.



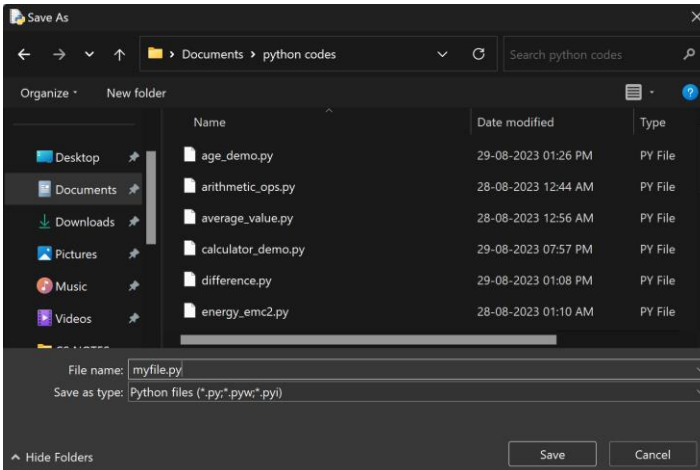
```
*untitled*
File Edit Format Run Options Window Help
num1 = 50
num2 = 6

result = num1 / num2
print(result)|
```

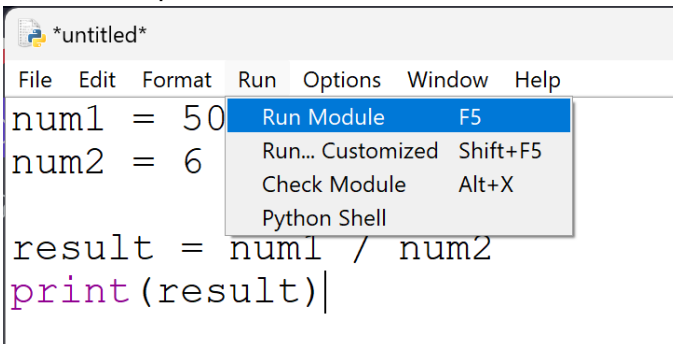
4. Save your Python Script by going to **File > Save**, and give any valid file name to your script with the extension **.py**



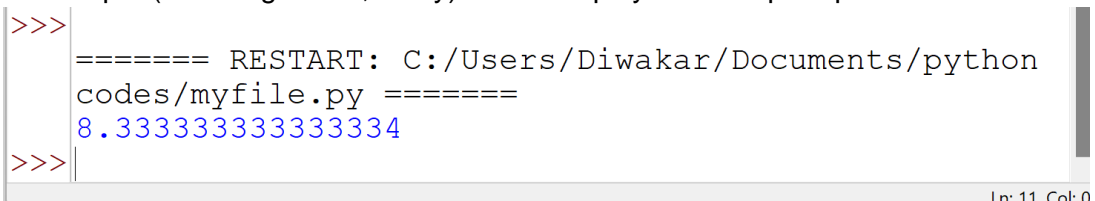
5. Click **Save** to save your script at a location of your choice.



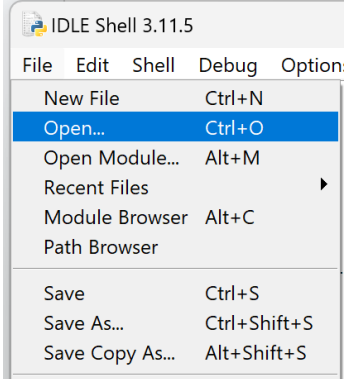
6. Once in script mode, choose **Run > Run Module** or press **F5** shortcut to run your script.



7. The output (including errors, if any) will be displayed in the prompt window.



8. To Open an existing Python script, choose **File > Open** and double click on the python file to Open it in IDLE Script Mode.



## Appendix - 3

### Working with Pydroid

Pydroid is a Python Interpreter & IDE on Google Play Store to write and run Python code on the Android device.

1. Go to Google Play Store on your Android Device. **Search for 'Pydroid' and Install the app.**



2. **Open** the Pydroid App.

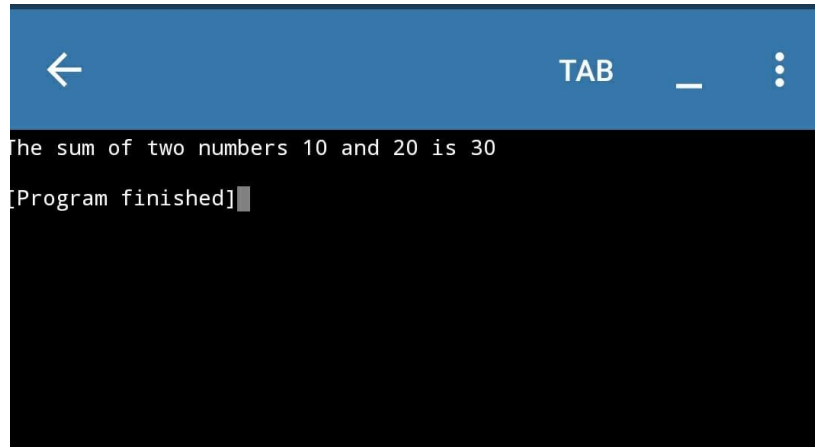
#### Using Script Mode

1. **Write Python code** in the built in code editor or open the existing ones.

```
new*  
1 a=10  
2 b=20  
3 sum=a+b  
4 print("The sum of two numbers",a,"and",b,"is",  
sum)
```



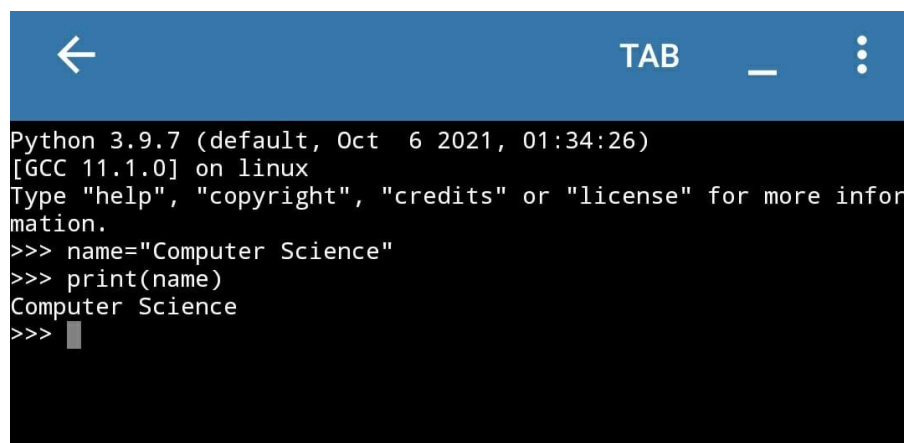
2. Write code in the script mode. Run the Python Code by tapping the **button (▶)**. Save the code.
3. The output is displayed on the screen.



```
← TAB _ ⋮  
The sum of two numbers 10 and 20 is 30  
[Program finished]
```

## Using Interactive Mode

1. `>>>` shows the **prompt** where Python statements can be written in the Interactive Mode.
2. The output will be displayed after each statement.



```
← TAB _ ⋮  
Python 3.9.7 (default, Oct 6 2021, 01:34:26)  
[GCC 11.1.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> name="Computer Science"  
>>> print(name)  
Computer Science  
>>> █
```

## Using Pip

1. Search for libraries and install or use prebuilt libraries repositories.
2. Quick Install option is also available.



**Class: XI**  
**Computer Science (083)**  
**Question Paper (Theory)**

**Time allowed: 3 Hours**

**Maximum Marks: 70**

**General Instructions:**

Please check this question paper which contains 35 questions.

- The paper is divided into 5 Sections- A, B, C, D and E.
- Section A, consists of 18 questions (1 to 18). Each question carries 1 Mark.
- Section B, consists of 7 questions (19 to 25). Each question carries 2 Marks.
- Section C, consists of 5 questions (26 to 30). Each question carries 3 Marks.
- Section D, consists of 2 questions (31 to 32). Each question carries 4 Marks.
- Section E, consists of 3 questions (33 to 35). Each question carries 5 Marks.
- All programming questions are to be answered using Python Language only.

<b>Q No.</b>	<b>Section-A (18 Marks)</b>	<b>Marks</b>
1	Python, a computer programming language is: a. Low-level language b. Machine level language c. Assembly level language d. High-level language	1
2	The default value of sep in print() is: a. Tab b. Space c. Percentage d. Newline	1
3	Which of the following is categorized as digital footprint? a. IP Address b. Email Address c. Both of the above d. None of the above	1

4	<p>What shape represents a decision in a flow chart?</p> <p>a. Arrow</p> <p>b. Oval</p> <p>c. Diamond</p> <p>d. Rectangle</p>	1
5	<p>State true or false.</p> <p>"A single element tuple is defined as t1=(5)."</p>	1
6	<p>What is the primary function of the Operating System?</p> <p>a. Managing hardware resources and providing user interface</p> <p>b. Controlling User's Actions</p> <p>c. Storing Data</p> <p>d. Browsing the Internet</p>	1
7	<p>Which of the following is an example of cyber crime?</p> <p>a. Ethical Hacking</p> <p>b. Sharing Files Over Internet</p> <p>c. Identity Theft</p> <p>d. Browsing Social Media</p>	1
8	<p>State true or false:</p> <p>A=4 and 4=A are interchangeably used in Python.</p>	1
9	<p>Which of the following is not a memory unit?</p> <p>a. RAM</p> <p>b. ROM</p> <p>c. Register</p> <p>d. Data Bus</p>	1
10	<p>Which of the following is used to delete an element from a dictionary?</p> <p>a. Pop()</p> <p>b. Remove()</p> <p>c. Delete()</p> <p>d. All of the above</p>	1

11	<p>Which operator is used for floor division?</p> <p>a. /</p> <p>b. //</p> <p>c. %</p> <p>d. None of the above</p>	1
12	<p>What will be the output for the following code?</p> <pre>y=7 z=3 y//=z print(y)</pre> <p>a. 2</p> <p>b. 3</p> <p>c. 4</p> <p>d. None of the above</p>	1
13	<p>Which of the following is no a type of error in Python?</p> <p>a. KeyError</p> <p>b. ValueError</p> <p>c. ZeroDivisionError</p> <p>d. FloatingError</p>	1
14	<p>A netizen must follow:</p> <p>a. Social Media Ettiquetes</p> <p>b. Communication Ettiquetes</p> <p>c. Net Etiquettes</p> <p>d. All of the above</p>	1
15	<p>Which keyword is used to stop the loop before it has looped through all the items?</p> <p>a. continue</p> <p>b. stop</p> <p>c. break</p> <p>d. exit</p>	1

16	<p>What will be the output?</p> <pre>a = 33 if a:     print(a)</pre> <p>a. False b. True c. a d. 33</p>	1
	<p>Q17 and 18 are ASSERTION AND REASONING based questions. Mark the correct choice as</p> <p>(a) Both A and R are true and R is the correct explanation for A (b) Both A and R are true and R is not the correct explanation for A (c) A is True but R is False (d) A is false but R is True</p>	
17	<p>Assertion(A): Copyright, trademark, and patent are forms of intellectual property protection. Reason(R): To safeguard the innovation or creative work from duplicacy, copyright, trademark and patent are granted to the creators.</p>	1
18	<p>Assertion(A): Python has a set of built in methods that you can use on strings. Reason(R): All string methods return new values. They do not change the original string.</p>	1
<b>Q No.</b>	<b>Section-B (14 Marks)</b>	<b>Marks</b>
19	<p>Identify and correct the errors in the given code:</p> <pre>print("Hello, World")     x = 5     y = 3 result = x + y print(result</pre> <p style="text-align: center;"><b>OR</b></p>	2

	<p>Identify and correct the errors in the given code:</p> <pre>name = "Abhay" age == 30 print("My name is", name, "and I am", age, "years old")</pre>	
20	<p>What is the difference between Primary Memory and Secondary Memory?</p>	2
21	<p>Predict the output of the Python code given below:</p> <pre>n = 10 sum_even = 0 for i in range(1, n + 1):     if i % 2 == 0:         sum_even += i print(sum_even)</pre> <p style="text-align: center;"><b>OR</b></p> <p>Predict the output of the Python code given below:</p> <pre>n = 5 factorial = 1 for i in range(1, n + 1):     factorial *= i print(factorial)</pre>	1+1=2
22	<p>What is the IT ACT?</p>	2
23	<p>What is the difference between Concatenation and Repetition Operators? Give example.</p> <p style="text-align: center;"><b>OR</b></p> <p>What are the different methods to add elements to a list? Give example.</p>	2
24	<p>Explain Net Etiquettes.</p> <p style="text-align: center;"><b>OR</b></p> <p>What are the key communication etiquettes that individuals should follow in the digital world to ensure effective online interaction?</p>	2

25	What practices can be used to ensure a safe and secure Web browsing?	2
<b>Q No</b>	<b>Section-C (15 Marks)</b>	<b>Marks</b>
26	Write a Python Program to take a input number 'n' from the user and print its square and cube.	3
27	Predict the output: <pre>n = 10 for i in range(n, 0, -1):     if i % 2 != 0:         print(i, end=' ', sep='--')     else:         print(i, end=' ', sep='++')</pre>	3
28	Write a program to calculate and display the sum of all odd numbers in a list.  <b>OR</b> Write a program to input 'n' numbers in a list, and find the minimum and maximum number among 'n' numbers.	3
29	What is a key value pair with reference to Python Dictionary? Explain with syntax and examples.	3
30	Write a Python Program to accept a string and display the word which has vowel 'a' in it.	3
<b>Q No.</b>	<b>Section-D (8 Marks)</b>	<b>Marks</b>
31	Write a program to accept a string from the user and display the same string without space.	4
32	Write a program to search for a given element from a list and display its index value.	4



Q No.	Section-E (15 Marks)	Marks
33	<p>Suggest appropriate functions for the following tasks:</p> <ol style="list-style-type: none"> <li>To check whether a string contain digits</li> <li>To check whether all the letters of the string are in uppercase</li> <li>To add a list to an existing list</li> <li>To delete existing element at a particular index from a list</li> <li>To get all the values of a dictionary for the existing keys</li> </ol>	5
34	<p>With reference to the above logic circuit, choose the best possible option for the following questions:</p> <ol style="list-style-type: none"> <li>What is the expression of R?</li> <li>What is the expression of Q?</li> <li>What is the expression of S?</li> <li>What is the expression of P?</li> <li>The truth table representation of the above logic will have _____ no. of rows.</li> </ol>	5
35	<p>Write a short note on the following:</p> <ol style="list-style-type: none"> <li>E-Waste Management</li> <li>Intellectual Property Rights</li> </ol>	5