DIRECTORATE OF EDUCATION
GNCT of Delhi

SUPPORT MATERIAL
(2024-2025)
Class : XI

COMPUTER SCIENCE

Under the Guidance of

Sh. Ashok Kumar
Secretary (Education)

Sh. R.N. Sharma
Director (Education)

Dr. Rita Sharma
Addl. DE (School & Exam.)

Coordinators

Mr. Sanjay Subhas Kumar
DDE (Exam)

Ms. Ritu Singhal
OSD (Exam)

Dr. Raj Kumar
OSD (Exam)

Mr. Krishan Kumar
OSD (Exam)

Production Team

Anil Kumar Sharma

Published at Delhi Bureau of Text Books, 25/2 Institutional Area, Pankha Road, New Delhi-110058 by **Rajesh Kumar,** Secretary, Delhi Bureau of Text Books and Printed by Supreme Offset Press, Greater Noida, U.P.

ASHOK KUMAR
IAS

सचिव (शिक्षा)
राष्ट्रीय राजधानी क्षेत्र
दिल्ली सरकार
पुराना सचिवालय, दिल्ली-110054
दूरभाष: 23890187 टेलीफैक्स : 23890119

Secretary (Education)
Government of National Capital Territory of Delhi
Old Secretariat , Delhi-110054
Phone : 23890187, Telefax : 23890119
E-mail : secyedu@nic.in

DE.5(228)(Exam)Message/SM/2018/555
Dated: 01/07/2024

MESSAGE

In the profound words of Dr. Sarvepalli Radhakrishnan, **"The true teachers are those who help us think for ourselves."**

Every year, our teams of subject experts shoulder the responsibility of updating the Support Material to synchronize it with the latest changes introduced by CBSE. This continuous effort is aimed at empowering students with innovative approaches and techniques, thereby fostering their problem-solving skills and critical thinking abilities.

I am confident that this year will be no exception, and the Support Material will greatly contribute to our students' academic success.

The development of the support material is a testament to the unwavering dedication of our team of subject experts. It has been designed with the firm belief that its thoughtful and intelligent utilization will undoubtedly elevate the standards of learning and continue to empower our students to excel in their examinations.

I wish to extend my heartfelt congratulations to the entire team for their invaluable contribution in creating this immensely helpful resource for our students.

Wishing all our students a promising and bright future brimming with success.

(ASHOK KUMAR)

R.N. SHARMA, IAS
Director, Education & Sports

Directorate of Education
Govt. of NCT of Delhi
Room No. 12, Old Secretariat
Near Vidhan Sabha,
Delhi-110054
Ph.: 011-23890172
E-mail : diredu@nic.in

DE-5/228/Exam/Manual/2018/576
Dated 04/07/2018

MESSAGE

It brings me great pleasure to present the support material specifically designed for students of classes IX to XII by our dedicated team of subject experts. The Directorate of Education remains resolute in its commitment to empower educators and students alike, extending these invaluable resources at no cost to students attending Government and Government-Aided schools in Delhi.

The support material epitomizes a commendable endeavour towards harmonizing content with the latest CBSE patterns, serving as a facilitative tool for comprehending, acquiring and honing essential skills and competencies stipulated within the curriculum.

Embedded within this initiative is a structured framework conducive to nurturing an analytical approach to learning and problem-solving. It is intended to prompt educators to reflect upon their pedagogical methodologies, forging an interactive conduit between students and academic content.

In the insightful words of Rabindranath Tagore, **"Don't limit a child to your own learning, for he was born in another time."**

Every child is unique, with their own interests, abilities and potential. By allowing children to learn beyond the scope of our own experiences, we support their individual growth and development, helping them to reach their full potential in their own right.

May every student embrace the joy of learning and be empowered with the tools and confidence to navigate and shape the future.

(R. N. SHARMA)

Dr. RITA SHARMA
Additional Director of Education
(School/Exam)

Govt. of NCT of Delhi
Directorate of Education
Old Secretariat, Delhi-110054
Ph.: 23890185

D.O. No DE.5/228/Exam/Message/SN/2018/570
Dated: 02/07/2024

MESSAGE

"Children are not things to be molded, but are people to be unfolded." - Jess Lair

In line with this insightful quote, the Directorate of Education, Delhi, has always made persistent efforts to nurture and unfold the inherent potential within each student. This support material is a testimony to this commitment.

The support material serves as a comprehensive tool to facilitate a deeper understanding of the curriculum. It is crafted to help students not only grasp essential concepts but also apply them effectively in their examinations. We believe that the thoughtful and intelligent utilization of these resources will significantly enhance the learning experience and academic performance of our students.

Our expert faculty members have dedicated themselves to the support material to reflect the latest CBSE guidelines and changes. This continuous effort aims to empower students with innovative approaches, fostering their problem-solving skills and critical thinking abilities.

I extend my heartfelt congratulations to the entire team for their invaluable contribution to creating a highly beneficial and practical support material. Their commitment to excellence ensures that our students are well-prepared to meet the challenges of the CBSE examinations and beyond.

Wishing you all success and fulfilment in your educational journey.

(Dr. Rita Sharma)

DIRECTORATE OF EDUCATION
Govt. of NCT, Delhi

SUPPORT MATERIAL
(2024-2025)
Class : XI

COMPUTER SCIENCE

NOT FOR SALE

PUBLISHED BY : DELHI BUREAU OF TEXTBOOKS

भारत का संविधान

उद्देशिका

हम, भारत के लोग, भारत को एक [1][**संपूर्ण प्रभुत्व-संपन्न समाजवादी पंथनिरपेक्ष लोकतंत्रात्मक गणराज्य**] बनाने के लिए, तथा उसके समस्त नागरिकों को :

सामाजिक, आर्थिक और राजनैतिक **न्याय**,

विचार, अभिव्यक्ति, विश्वास, धर्म

और उपासना की **स्वतंत्रता**,

प्रतिष्ठा और अवसर की **समता**

प्राप्त कराने के लिए,

तथा उन **सब में**

व्यक्ति की गरिमा और [2][**राष्ट्र की एकता और अखंडता**] सुनिश्चित करने वाली **बंधुता**

बढ़ाने के लिए

दृढ़संकल्प होकर **अपनी इस संविधान सभा में** आज तारीख 26 नवंबर, 1949 ई. को **एतद्द्वारा इस संविधान को अंगीकृत, अधिनियमित और आत्मार्पित करते हैं।**

---

1. संविधान (बयालीसवां संशोधन) अधिनियम, 1976 की धारा 2 द्वारा (3.1.1977 से) "प्रभुत्व-संपन्न लोकतंत्रात्मक गणराज्य" के स्थान पर प्रतिस्थापित।

2. संविधान (बयालीसवां संशोधन) अधिनियम, 1976 की धारा 2 द्वारा (3.1.1977 से) "राष्ट्र की एकता" के स्थान पर प्रतिस्थापित।

THE CONSTITUTION OF INDIA

PREAMBLE

WE, THE PEOPLE OF INDIA, having solemnly resolved to constitute India into a [1][**SOVEREIGN SOCIALIST SECULAR DEMOCRATIC REPUBLIC**] and to secure to all its citizens :

JUSTICE, social, economic and political;

LIBERTY of thought, expression, belief, faith and worship;

EQUALITY of status and of opportunity; and to promote among them all

FRATERNITY assuring the dignity of the individual and the [2][unity and integrity of the Nation];

IN OUR CONSTITUENT ASSEMBLY this twenty-sixth day of November, 1949 do **HEREBY ADOPT, ENACT AND GIVE TO OURSELVES THIS CONSTITUTION.**

---

1. Subs. by the Constitution (Forty-second Amendment) Act, 1976, Sec.2, for "Sovereign Democratic Republic" (w.e.f. 3.1.1977)
2. Subs. by the Constitution (Forty-second Amendment) Act, 1976, Sec.2, for "Unity of the Nation" (w.e.f. 3.1.1977)

भारत का संविधान

भाग 4क

नागरिकों के मूल कर्तव्य

अनुच्छेद 51 क

मूल कर्तव्य - भारत के प्रत्येक नागरिक का यह कर्तव्य होगा कि वह -

(क) संविधान का पालन करे और उसके आदर्शों, संस्थाओं, राष्ट्रध्वज और राष्ट्रगान का आदर करे;

(ख) स्वतंत्रता के लिए हमारे राष्ट्रीय आंदोलन को प्रेरित करने वाले उच्च आदर्शों को हृदय में संजोए रखे और उनका पालन करे;

(ग) भारत की संप्रभुता, एकता और अखंडता की रक्षा करे और उसे अक्षुण्ण बनाए रखे;

(घ) देश की रक्षा करे और आह्वान किए जाने पर राष्ट्र की सेवा करे;

(ङ) भारत के सभी लोगों में समरसता और समान भ्रातृत्व की भावना का निर्माण करे जो धर्म, भाषा और प्रदेश या वर्ग पर आधारित सभी भेदभावों से परे हो, ऐसी प्रथाओं का त्याग करे जो महिलाओं के सम्मान के विरुद्ध हों;

(च) हमारी सामासिक संस्कृति की गौरवशाली परंपरा का महत्त्व समझे और उसका परिरक्षण करे;

(छ) प्राकृतिक पर्यावरण की, जिसके अंतर्गत वन, झील, नदी और वन्य जीव हैं, रक्षा करे और उसका संवर्धन करे तथा प्राणिमात्र के प्रति दयाभाव रखे;

(ज) वैज्ञानिक दृष्टिकोण, मानववाद और ज्ञानार्जन तथा सुधार की भावना का विकास करे;

(झ) सार्वजनिक संपत्ति को सुरक्षित रखे और हिंसा से दूर रहे;

(ञ) व्यक्तिगत और सामूहिक गतिविधियों के सभी क्षेत्रों में उत्कर्ष की ओर बढ़ने का सतत् प्रयास करे, जिससे राष्ट्र निरंतर बढ़ते हुए प्रयत्न और उपलब्धि की नई ऊँचाइयों को छू सके; और

(ट) यदि माता-पिता या संरक्षक है, छह वर्ष से चौदह वर्ष तक की आयु वाले अपने, यथास्थिति, बालक या प्रतिपाल्य को शिक्षा के अवसर प्रदान करे।

Constitution of India

Part IV A (Article 51 A)

Fundamental Duties

It shall be the duty of every citizen of India —

(a) to abide by the Constitution and respect its ideals and institutions, the National Flag and the National Anthem;

(b) to cherish and follow the noble ideals which inspired our national struggle for freedom;

(c) to uphold and protect the sovereignty, unity and integrity of India;

(d) to defend the country and render national service when called upon to do so;

(e) to promote harmony and the spirit of common brotherhood amongst all the people of India transcending religious, linguistic and regional or sectional diversities; to renounce practices derogatory to the dignity of women;

(f) to value and preserve the rich heritage of our composite culture;

(g) to protect and improve the natural environment including forests, lakes, rivers, wildlife and to have compassion for living creatures;

(h) to develop the scientific temper, humanism and the spirit of inquiry and reform;

(i) to safeguard public property and to abjure violence;

(j) to strive towards excellence in all spheres of individual and collective activity so that the nation constantly rises to higher levels of endeavour and achievement;

* (k) who is a parent or guardian, to provide opportunities for education to his child or, as the case may be, ward between the age of six and fourteen years.

Note: The Article 51A containing Fundamental Duties was inserted by the Constitution (42nd Amendment) Act, 1976 (with effect from 3 January 1977).

*(k) was inserted by the Constitution (86th Amendment) Act, 2002 (with effect from 1 April 2010).

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

CLASS – XI (2024-25)

SUBJECT: COMPUTER SCIENCE

S. No.	Name & Designation	Name of School Branch
1.	MR. TRILOK CHAND HALDUNIA (PRINCIPAL)	SBV, TUGHLAKABAD EXTENSION, SCHOOL ID: 1925056
2.	MR. PRASHANT KUMAR SINGH (LECTURER COMPUTER SCIENCE)	RSBV KIRAN VIHAR, DELHI - 92, SCHOOL ID – 1001109
3.	MR. SACHIN KUMAR (LECTURER COMPUTER SCIENCE)	BURARI - SBV, DELHI – 84 SCHOOL ID - 1207009
4.	MR. ANIRUDRA DIWAKAR (LECTURER COMPUTER SCIENCE)	SBV SEEMAPURI, DELHI – 95 SCHOOL ID - 1106009
5.	MS. YAMINI SHARMA (LECTURER COMPUTER SCIENCE)	SKV, H-BLOCK, PH-I, ASHOK VIHAR,SCHOOL ID - 1411026
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

Syllabus

Computer Science (2024-25)

CLASS XI Code No. 083

1. Learning Outcomes

Students should be able to:

(a) develop basic computational thinking

(b) explain and use data types

(c) appreciate the notion of algorithms

(d) develop a basic understanding of computer systems architecture and operating system

(e) explain cyber ethics, cyber safety, and cybercrime

(f) understand the value of technology in societies along with consideration of gender and disability issues.

2. Distribution of Marks

Unit No.	Unit Name	Marks	Periods	
			Theory	Practical
1.	Computer Systems and Organisation	10	10	10
2.	Computational Thinking and Programming -1	45	80	60
3.	Society, Law, and Ethics	15	20	—
	Total	**70**	**110**	**70**

3. Unit wise Syllabus

Unit 1: Computer Systems and Organisation

- Basic computer organisation: Introduction to Computer System, hardware, software, input device, output device, CPU, memory (primary, cache and secondary), units of memory (bit, byte, KB, MB, GB, TB, PB)
- Types of software: System software (Operating systems, system utilities, device drivers), programming tools and language translators (assembler, compiler, and interpreter), application software
- Operating System(OS): functions of the operating system, OS user interface
- Boolean logic: NOT, AND, OR, NAND, NOR, XOR, NOT, truth tables and De Morgan's laws, Logic circuits

- Number System: Binary, Octal, Decimal and Hexadecimal number system conversion between number systems
- Encoding Schemes: ASCII, ISCII, and Unicode (UTF8, UTF32)

Unit 2: Computational Thinking and Programming - 1

- **Introduction to Problem-solving:** Steps for Problem-solving (Analyzing the problem, developing an algorithm, coding, testing, and debugging), representation of algorithms using flowchart and pseudocode, decomposition
- Familiarization with the basics of Python programming: Introduction to Python, Features of Python, executing a simple "hello world" program, execution modes: interactive mode and script mode, Python character set, Python tokens (keyword, identifier, literal, operator, punctuator), variables, concept of l-value and r-value, use of comments
- **Knowledge of data types:** Number(integer, floating point, complex), boolean, sequence (string, list, tuple), None, Mapping(dictionary), mutable and immutable data types.
- **Operators:** arithmetic operators, relational operators, logical operators, assignment operators, augmented assignment operators, identity operators (is, is not), membership operators (in not in)
- Expressions, statement, type conversion, and input/output: precedence of operators, expression, evaluation of an expression, type-conversion (explicit and implicit conversion), accepting data as input from the console and displaying output.
- Errors-syntax errors, logical errors, and run-time errors
- **Flow of Control:** introduction, use of indentation, sequential flow, conditional and iterative flow
- **Conditional statements:** if, if-else, if-elif-else, flowcharts, simple programs: e.g.: absolute value, sort 3 numbers and divisibility of a number.
- **Iterative Statement:** for loop, range(), while loop, flowcharts, break and continue statements, nested loops, suggested programs: generating pattern, summation of series, finding the factorial of a positive number, etc.
- **Strings:** introduction, string operations (concatenation, repetition, membership and slicing), traversing a string using loops, built-in functions/methods-len(), capitalize(), title(), lower(), upper(), count(), find(), index(), endswith(), startswith(), isalnum(), isalpha(), isdigit(), islower(), isupper(), isspace(), lstrip(), rstrip(), strip(), replace(), join(), partition(), split()

- **Lists:** introduction, indexing, list operations (concatenation, repetition, membership and slicing), traversing a list using loops, built-in functions/ methods-len(), list(), append(), extend(), insert(), count(), index(), remove(), pop(), reverse(), sort(), sorted(), min(), max(), sum(); nested lists, suggested programs: finding the maximum, minimum, mean of numeric values stored in a list; linear search on list of numbers and counting the frequency of elements in a list.
- **Tuples:** introduction, indexing, tuple operations (concatenation, repetition, membership and slicing); built-in functions/methods len(), tuple(), count(), index(), sorted(), min(), max(), sum(); tuple assignment, nested tuple; suggested programs: finding the minimum, maximum, mean of values stored in a tuple; linear search on a tuple of numbers, counting the frequency of elements in a tuple.
- **Dictionary:** introduction, accessing items in a dictionary using keys, mutability of a dictionary (adding a new term, modifying an existing item), traversing a dictionary, built-in functions/methods len(), dict(), keys(), values(), items(), get(), update(), del, clear(), fromkeys(), copy(), pop(), popitem(), setdefault(), max(), min(), sorted(); Suggested programs: count the number of times a character appears in a given string using a dictionary, create a dictionary with names of employees, their salary and access them.
- **Introduction to Python modules:** Importing module using 'import <module>' and using from statement, importing math module (pi, e, sqrt(), ceil(), floor(), pow(), fabs(), sin(), cos(), tan()); random module (random(), randint(), randrange()), statistics module (mean(), median(), mode()).

Unit 3: Society, Law and Ethics

- Digital Footprints
- Digital Society and Netizen: net etiquettes, communication etiquettes, social media etiquettes
- Data Protection: Intellectual property rights (copyright, patent, trademark), violation of IPR (plagiarism, copyright infringement, trademark infringement), open source software and licensing (Creative Commons, GPL and Apache)
- Cyber Crime: definition, hacking, eavesdropping, phishing and fraud emails, ransomware, cyber trolls, cyber bullying
- Cyber safety: safely browsing the web, identity protection, confidentiality

- Malware: viruses, trojans, adware
- E-waste management: proper disposal of used electronic gadgets.
- Information Technology Act (IT Act)
- Technology and society: Gender and disability issues while teaching and using computers

4. Practical

S.No.	Unit Name	Marks (Total=30)
1.	**Lab Test (12 marks)**	
	Python program (60% logic + 20% documentation + 20% code quality)	**12**
2.	**Report File + Viva (10 marks)**	
	Report file: Minimum 20 Python programs	**7**
	Viva voce	**3**
	Project (that uses most of the concepts that have been learnt	**8**

5. Suggested Practical List

Python Programming

- Input a welcome message and display it.
- Input two numbers and display the larger / smaller number. Input three numbers and display the largest / smallest number.
- Generate the following patterns using nested loops:

Pattern-1	Pattern-2	Pattern-3
*	12345	A
**	1234	AB
***	123	ABC
****	12	ABCD
*****	1	ABCDE

- Write a program to input the value of x and n and print the sum of the following series:
  - $1 + x - x^2 + x^3 + x^4 + \ldots x^n$
  - $1 - x + x^2 - x^3 + x^4 - \ldots x^n$

- ❖ $x+\frac{x^2}{2}+\frac{x^3}{3}+\frac{x^4}{4}+....\frac{x^n}{n}$
- ❖ $x+\frac{x^2}{2!}+\frac{x^3}{3!}+\frac{x^4}{4!}+....\frac{x^n}{n!}$

- Determine whether a number is a perfect number, an Armstrong number or a palindrome.
- Input a number and check if the number is a prime or composite number.
- Display the terms of a Fibonacci series.
- Compute the greatest common divisor and least common multiple of two ntegers.
- Count and display the number of vowels, consonants, uppercase, lowercase characters in string.
- Input a string and determine whether it is a palindrome or not; convert the case of characters in a string.
- Find the largest/smallest number in a list/tuple
- Input a list of numbers and swap elements at the even location with the elements at the odd location.
- Input a list/tuple of elements, search for a given element in the list/tuple.
- Create a dictionary with the roll number, name and marks of n students in a class and display the names of students who have marks above 75.

6. Suggested Reading Material

- NCERT Textbook for Computer Science (Class XI)
- Support Material on CBSE website

Computer Systems and Organisation

1.1 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. Computer is a **combination of hardware and software**. Computer systems work on the **IPO (Input-Process-Output)** Model. They process information stored in **binary** form **(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, memory addresses and control information**.
- **Software is a set of instructions**, 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 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 stored resources
- **Storage** - can store large amount 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

Input Devices

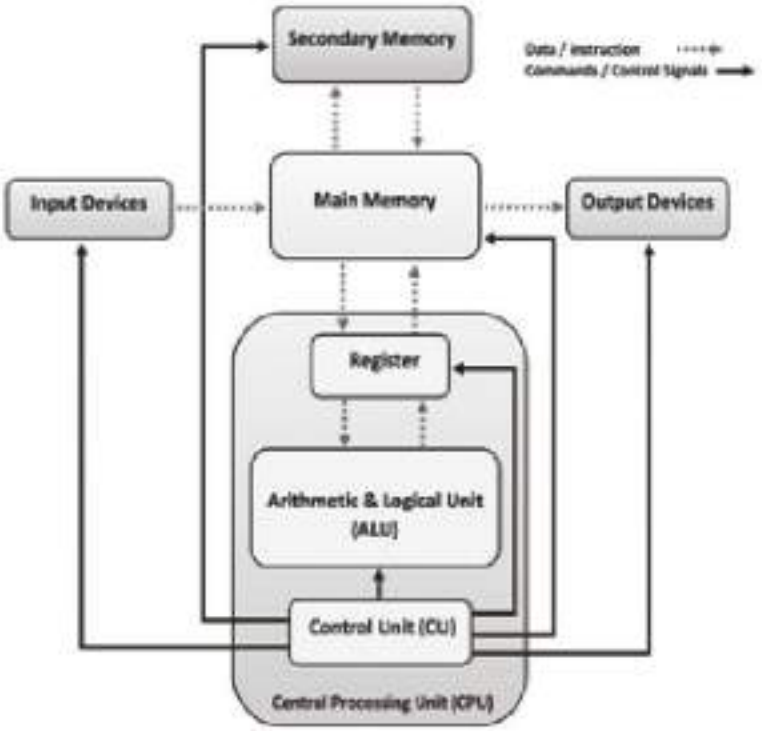
- The hardware devices that **send input data from the user to the CPU** are known as Input Devices.
- They are responsible for capturing, and transferring data from the outside world into the computer system.
- Some of the popular Input Devices are Keyboard, Mouse, Joystick, Scanner, Microphone, Optical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

Output Devices

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

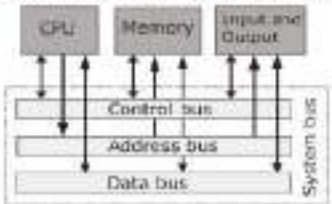
Central Processing Unit

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 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 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.

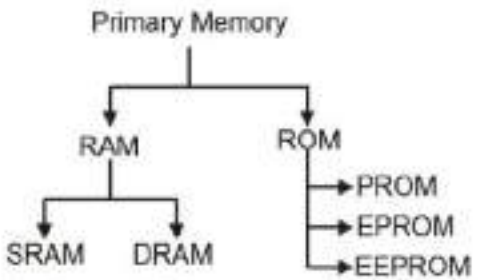


Memory

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.
3. 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.
4. To reduce this waiting time, and increase the throughput of the CPU, a cache memory (buffer) is placed between the CPU and primary memory.
5. The cache holds frequently used instructions and data and makes them readily available to the CPU.

Secondary Memory

1. Secondary memory is also known as **permanent** memory.
2. It is **non-volatile** in nature, i.e. the data **does not get lost even after power is turned off**.
3. Examples of Secondary memory are Hard Disk Drive, Solid State Drive, USB Flash Drive, Floppy Drive, Compact Disk (CD), Digital Versatile Disk (DVD).
4. The CPU does not interact with the Secondary Memory directly.

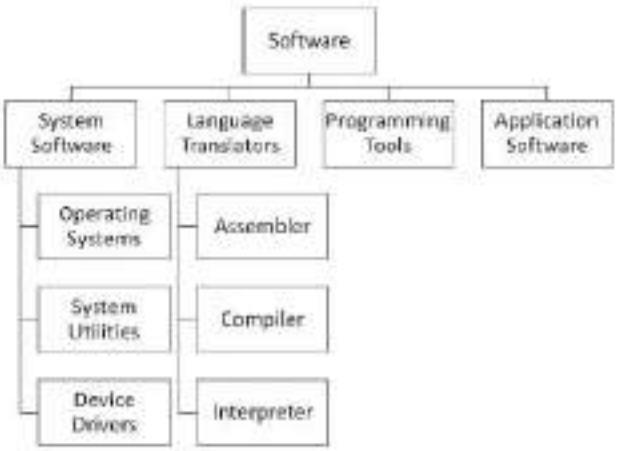
Units of Memory

1. The smallest unit of memory is called a binary digit, or a bit.
2. The bit can assume only two values, 0 and 1.

3. A sequence of 4-bits is called a Nibble.
4. 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

1.2 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 numbers. Once all errors are corrected and object code is created, the 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 the same to move forward.
- Assembly language code can be converted to machine code (Binary Code) using a translator called Assembler.

Programming Tools

- Tools that assist the users/developers in creating, editing, testing and debugging during the development of software are known as Programming Tools.
- It includes development tools, code editors and translators.
- Examples include: Visual Studio Code, Sublime Text, Eclipse etc.

1.3 OPERATING SYSTEMS (OS)

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 Systems

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 Systems

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

2. Mobile Operating Systems
   (a) Google Android
   (b) Apple iOS
   (c) Symbian OS
   (d) Microsoft Windows RT OS

Functions of Operating System

1. **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

2. **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

3. **Process Management:** It manages all the programs in execution (processes).

   It includes:

   - Process Creation & Termination
   - Process Scheduling
   - Context Switching

4. **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
5. **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
6. **Security:** The operating system protects the system from all the threats, unauthorized access and other vulnerabilities.

   It includes:
   - Authentication
   - Authorization
   - Data Encryption
   - Firewall Management
7. **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 (UI)

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

1. **Command Line Interface:** In this interface the user interacts with the system through typed commands. Every operation is carried out on the basis of commands entered by the user on a terminal, also known as command prompt.
2. **Graphical Based User Interface:** In this interface users interact with the system through graphical items such as icons, menu, taskbar etc.
3. **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.

4. **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.
5. **Touch Based User Interface:** Touch based user interface requires a physical touch through the input device.

1.4 BOOLEAN LOGIC

Boolean Logic is a concept that involves binary variables and operations. The binary variables can assume only two values, namely, 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 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 a Boolean value of 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 one 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 operand is False, and returns False (0) when the operand is True (1).

Example: A' / $\bar{A}$ / ~A may be read as A complement or Not of A.

Truth Tables

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

The number of rows in a truth table are 2^n, where n is number of input variables.

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

Truth Table of the function F = A OR B

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'

DeMorgan'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.

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

Truth Table for DeMorgan's First Law

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.

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

Truth Table for DeMorgan'sSecond Law

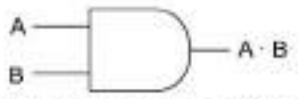
A	B	A'	B'	A+B	(A+B)'	A'.B'
0	0	1	1	0	**1**	**1**
0	1	1	0	1	**0**	**0**
1	0	0	1	1	**0**	**0**
1	1	0	0	1	**0**	**0**

Logic Gates

A logic gate is a fundamental building block of digital circuits. It performs a Boolean Function which takes one or more inputs in the form of 0 or 1 and generates a particular output which is governed by 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 for AND Gate

Truth Table for AND Gate

A	B	A · B
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**

OR Gate

Logic Gate Diagram for OR Gate

Truth Table for OR Gate

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

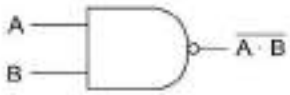
NOT Gate

Logic Gate Diagram for NOT Gate

Truth Table for NOT Gate

A	~A
0	**1**
1	**0**

NAND Gate



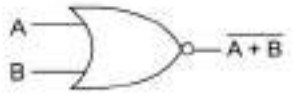
Logic Gate Diagram for NAND Gate

Truth Table for NAND Gate

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 for NOR Gate

Truth Table for NOR Gate

A	B	A + B	X = (A + B)′
0	0	0	**1**
0	1	1	**0**
1	0	1	**0**
1	1	1	**0**

XOR Gate

Logic Gate Diagram for XOR Gate

Truth Table for XOR Gate

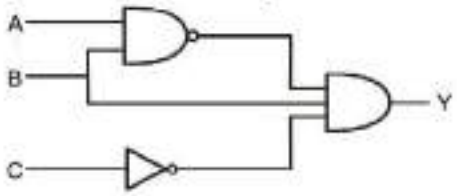
A	B	A ⊕ B
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**0**

Mathematically, the XOR operation $A \oplus B$ is equivalent to $AB' + A'B$

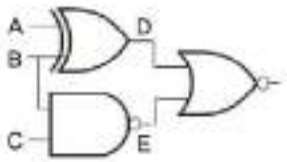
Logic Circuits

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:



Y=(A.B)'.B.C'



Y = ?

1.5 NUMBER SYSTEM

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

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

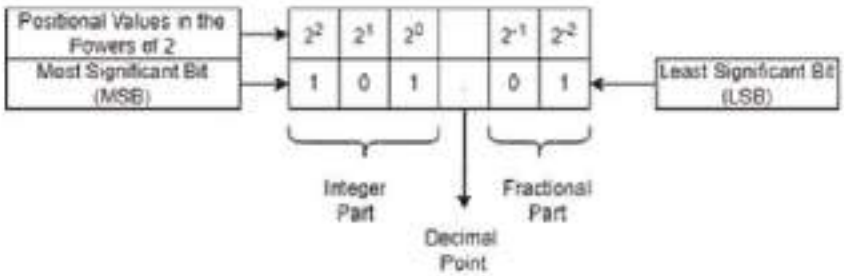
The powers of 10, which correspond respectively to the digits in a decimal integer when read from right to left, are called the place values of the digits.

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 1 and 0 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 used	No. of binary numbers possible
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 (Repeated Division/Multiplication Method)
   (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 (Positional Value Method)
   (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 (Repeated Division/Multiplication Method)

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).

Question-1. Convert decimal number 75 to binary number.

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

$=(1001011)_2$

Question-2. Convert decimal number 142.67 to a binary number.

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

.67 × 2 = 1.34	1	.34
.34 × 2 = 0.68	0	.68
.68 × 2 = 1.36	1	.36
.36 × 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).

Question-1. Convert decimal number 104 to octal number.

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

$= (150)_8$

Question-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	MSB

.93 × 8 = 7.44	7	.44
.44 × 8 = 3.52	3	.52
.52 × 8 = 4.16	4	.16
.16 × 8 = 1.28	1	.28

$= (1063.7341)_8$

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).

Question-1. Convert decimal number 600 to hexadecimal number.

Divisor	Dividend	Remainder	
16	600		
16	37	8	LSB
16	2	5	
	0	2	MSB

$= (258)_{16}$

Question-2. Convert decimal number 49 to hexadecimal number.

Divisor	Dividend	Remainder	
16	49		
16	3	1	LSB
	0	3	MSB

$= (31)_{16}$

Binary Number System to Decimal Number System Conversion

Steps:

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

Question-1. Convert Binary Number 101101 into Decimal Number.

101101

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

$= 32 + 0 + 8 + 4 + 0 + 1$

$= 45$

Question-2. Convert Binary Number 110101.010 into Decimal Number.

110101.010

$= 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3}$

$= 32 + 16 + 0 + 4 + 0 + 1 + 0 + 0.25 + 0$

$= 53.25$

Octal Number System to Decimal Number System Conversion

Steps:

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

Question-1. Convert Octal Number 156 into Decimal Number.

156

$= 1 \times 8^2 + 5 \times 8^1 + 6 \times 8^0$

$= 64 + 40 + 6$

$= 110$

Question-2. Convert Octal Number 76.12 into Decimal Number.

76.12

$= 7 \times 8^1 + 6 \times 8^0 + 1 \times 8^{-1} + 2 \times 8^{-2}$

$= 56 + 6 + 0.125 + 0.03125$

$= 62.15625$

Hexadecimal Number System to Decimal Number System Conversion

Steps:

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

Question-1. Convert Hexadecimal Number 0.87 into Decimal Number.

0.87

$= 0 \times 16^0 + 8 \times 16^{-1} + 7 \times 16^{-2}$

$= 0 + 0.5 + 0.02734375$

$= 0.52734375$

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 to complete the grouping of 4 bits.*

Question-1. Convert binary number 1000110101110 to hexadecimal number.

Converting Binary Number into Decimal Number:

$= (1000110101110)_2$

$= 1 \times 2^{12} + 0 \times 2^{11} + 0 \times 2^{10} + 0 \times 2^9 + 1 \times 2^8 + 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$

$= 4096 + 0 + 0 + 0 + 256 + 128 + 0 + 32 + 0 + 8 + 4 + 2 + 0$

$= (4526)_{10}$

Converting Decimal into Hexadecimal Number:

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

$= (11AE)_{16}$

Question-2. Convert Binary number 10010111101101 into Hexadecimal Number.

0010	0101	1110	1101
2	5	E	D

$= (25ED)_{16}$

Question-3. Convert Binary number 101100100.100111 into Hexadecimal Number.

0001	0110	0100	.	1001	1100
1	6	4		9	C

Hence the equivalent Hexadecimal Number is $(164.9C)_{16}$

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

Question-1. Convert Hexadecimal Number 7B316 to Binary Equivalent Number.

7	B	3	1	6
0111	1011	0011	0001	0110

Hence the equivalent binary number is $(01111011001100010110)_2$

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

D	2	.	9	2	A	B
1101	0010		1001	0010	1010	1011

Hence the equivalent binary number is $(11010010.100100101010101011)_2$

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 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 to complete the grouping of 3 bits.*

Question-1. Convert binary number 1011011010 to octal number.

$= (1332)_8$

001	011	011	010
1	**3**	**3**	**2**

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

Question-1. Convert Octal Number 767 to Binary Equivalent Number.

7	6	7
111	**110**	**111**
= 111	110	111

Hence the equivalent binary number is $(111110111)_2$

Question-2. Convert Octal Number 4716.15 to Binary Equivalent Number.

4	7	1	6	.	1	5
100	**111**	**001**	**110**		**001**	**101**

$= (100111001110.001101)_2$

Hexadecimal Number System to Octal Number System

Steps:

1. Convert each digit of the hexadecimal number into a 4-bit binary equivalent number.
2. Combine 3 bits from the LSB, and convert each group of 3 bits into octal using a table.

Question-1. Convert Hexadecimal Number 4B2A into Octal Number.

4	B (11)	2	A (10)
0100	1011	0010	1010

= 0100 1011 0010 1010

= 0 100 101 100 101 010

= 000 100 101 100 101 010

= $(45452)_8$

Question-2. Convert Hexadecimal Number FE6 into Octal Number.

F	E	6
1111	1110	0110

= 111 111 100 110

= $(7746)_8$

Octal Number System to Hexadecimal Number System

Steps:

1. Convert each digit of the octal number into a 3 bit binary equivalent number.
2. Combine 4 bits from the LSB, and convert each group of 4 bits into hexadecimal using a table.

Question-1. Convert Octal Number 2106 into Hexadecimal Number.

2	1	0	6
010	**001**	**000**	**110**

= 010 001 000 110

= 0100 0100 0110

= $(446)_{16}$

Question-2. Convert Octal Number 765 into Hexadecimal Number.

7	6	5
111	**110**	**101**

= 111 110 101

= 1 1111 0101

= 0001 1111 0101

= $(1F5)_{16}$

1.6 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.

SOLVED QUESTIONS AND ANSWERS

Fill in the Blanks

1. The binary language has two digits namely __0__ and __1__ .
2. The full form of the CPU is the Central Processing Unit.
3. The full form of ALU is the Arithmetic Logic Unit.
4. The full form of CU is the Control Unit.
5. Set of Registers are temporary and local storage for the CPU.
6. The System Bus provides a way for the CPU to communicate with the connected devices.
7. Encoding Schemes transforms text into number codes that facilitate the communication among computers.
8. Unicode is the encoding standard that is capable of representing multiple scripts.
9. The CPU is the brain of the computer responsible for executing instructions.
10. RAM stands for Random Access Memory.
11. The cache memory stores the most frequently used data and instructions to improve system performance.
12. The ROM (Read-Only Memory) is a non-volatile memory that retains its data even when the power is turned off.
13. The central processing unit (CPU) consists of the control unit and the arithmetic logic unit (ALU).
14. The primary purpose of an operating system is to manage computer resources.
15. A group of eight bits is known as a byte.
16. An instruction is a basic arithmetic or logical operation that the CPU can perform.
17. The physical components of a computer system are referred to as hardware.
18. A software is a collection of programs and data that enables a computer to perform a specific task.

State True or False

1. Microphone is an Input Device. (True)
2. Hardware cannot be seen or touched physically. (False)

3. Software cannot be seen or touched physically. (True)
4. Operating System is the interface between the User and the Hardware. (True)
5. MICR stands for Magnetic Ink Character Reader. (True)
6. ALU stands for Arithmetic and Logic Unit. (True)
7. Registers are permanent storage. (False)
8. 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)
9. RAM is categorized as DRAM and SRAM. (True)
10. EEPROM stands for Electronic Erasable Primary ROM. (False)
11. A sequence of 4-bits is called a byte. (False)
12. The compiler converts high level language into low level language at once. (True)
13. Gesture Based User Interface allows users to perform tasks without physically touching the system. (True)
14. Boolean OR Operator results True if both the inputs are true, otherwise false. (False)
15. A logic circuit is an electronic circuit used in computers to perform a logical operation using Logic Gates. (True)
16. The Control Unit is responsible for Arithmetic and Logical Operations. (False)
17. System Bus is a pathway for signals to travel from one location to another in the Computer System. (True)
18. The CPU directly interacts with the Secondary Memory. (False)
19. Typically, RAM is faster than Hard Disk Drives. (True)
20. RAM is larger in size than the Secondary Memory. (False)
21. Static RAM needs constant refreshing to preserve the data. (False)
22. RAM is non-volatile in nature. (False)
23. ROM stores the software that is used to start-up the computer. (True)
24. BIOS stands for Basic Input Output System. (True)
25. Cache memory reduces the access time of RAM by storing the frequently accessed items. (True)
26. The base of binary, octal, decimal and hexadecimal number systems are 2,8,10 and 16 respectively. (True)
27. A binary number is made of a sequence of bits. (True)
28. 1 KiloByte (KB) = 1024 MegaBytes (MB) (False)

29. Each hexadecimal digit has a unique 4-bit representation. (True)

30. 1 MegaByte (MB) = 1024 KiloBytes (KB) (True)

31. Interpreter is a System Utility software. (False)

32. Compiler is a Language Translator Software. (True)

33. Microsoft Windows is an example of Application Software. (False)

34. Device Drivers translate code from high level language to low level language. (False)

35. Compiler and Interpreter convert high level languages to low level languages. (True)

36. 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 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 NOR 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

(a) Both A and R are true and R is the correct explanation for A.

(b) Both A and R are true but R is not the correct explanation for A.

(c) A is True but R is False.

(d) A is False but R is True.

1. **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.

2. **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.

3. **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.

4. **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.

5. **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.

6. **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.

UNSOLVED QUESTIONS AND ANSWERS

Very Short Answer Questions

1. Explain the computer system.
2. What is meant by an Input Device?
3. What is meant by an Output Device?
4. Explain the organization of Computers.
5. What is Hardware?
6. What is Software?
7. What is the expansion (full form) of the CPU?
8. Is software tangible in nature?
9. Is hardware tangible in nature?
10. Which gate is also known as an inverter?

11. Which gate returns a true result if both inputs are true otherwise false?
12. With respect to Boolean logic, the dot operator (.) is used for _____ operation.

Short Answer Questions

1. What is computer organization?
2. What are the advantages of computers?
3. Name two common input devices.
4. Name two common output devices.
5. What are the types of memory?
6. What are examples of secondary storage devices?
7. What is cache memory?
8. What are the types of software?
9. What is system software?
10. What is an operating system (OS)?
11. What are system utilities?
12. What are device drivers?
13. What are language translators?
14. What is a compiler?
15. What is an assembler?
16. What is an interpreter?
17. What are programming tools?
18. How does system software differ from application software?
19. What role does an operating system play in a computer system?
20. Why are device drivers important for computer systems?
21. How does an interpreter differ from a compiler?
22. What are the functions of an operating system?
23. What is the significance of security in an operating system?
24. What is a command-line interface (CLI)?
25. What is a graphical user interface (GUI)?
26. What role does an operating system play in device management?
27. What is the first De Morgan's law?
28. What is the second De Morgan's law?
29. What is the function of an AND gate?

30. What is the function of an OR gate?
31. What is the function of a NOT gate?
32. What is the function of a NAND gate?
33. What is the function of a NOR gate?
34. What is the function of an XOR gate?
35. What are logic circuits?
36. What are the advantages of using hexadecimal notation?
37. What are encoding schemes?
38. What is ASCII?
39. What is ISCII?
40. What is Unicode?
41. What are the advantages of Unicode over ASCII?
42. Name two input devices commonly used for gaming.
43. What input device would you use to convert physical documents into digital format?
44. Draw the truth table for XOR and OR gate.
45. What output device would you use to produce audio from a computer?
46. Name and describe the function of the two main components within a CPU.
47. List any one difference between hardware and software.
48. Name two types of operations that the ALU performs.
49. Name any two hardware devices.
50. What are the major types of software?
51. How many megabytes (MB) are in a gigabyte (GB)?
52. If a disk drive has a capacity of 1 terabyte (TB), how many gigabytes (GB) does it have?
53. Convert 2,048 bytes to kilobytes (KB).
54. Which type of memory is commonly used for long-term storage of data and programs?
55. What is the main difference between primary memory and secondary memory?
56. Which type of memory is volatile, meaning it loses its contents when the power is turned off?
57. What is the purpose of cache memory in a computer system?
58. What is the purpose of general-purpose software?
59. Give an example of general-purpose software commonly used for word processing.

60. Name a popular general-purpose software used for creating presentations.

61. Give an example of a programming language that uses a compiler for translation.

62. What are universal gates? Why are they called universal gates?

Long Answer Questions

1. Explain the organization of computer systems.
2. What are input devices? List 5 input devices.
3. What are output devices? List 5 output devices.
4. What is the IPO Cycle?
5. Explain the functions of the Operating System.
6. Explain different types of OS User Interface.
7. Explain the difference between primary memory and secondary memory.
8. What is the need of Cache memory?
9. How are high level languages converted into low level languages?
10. Write down some advantages of computers.
11. Convert the following numbers from one number system to another:

(a) $(945)_{10} = (?)_2$ **Ans.** $(1110110001)_2$

(b) $(42)_{10} = (?)_8$ **Ans.** $(52)_8$

(c) $(194)_{10} = (?)_{16}$ **Ans.** $(C2)_{16}$

(d) $(10110)_2 = (?)_{10}$ **Ans.** $(22)_{10}$

(e) $(745)_8 = (?)_{10}$ **Ans.** $(485)_{10}$

(f) $(E1A)_{16} = (?)_{10}$ **Ans.** $(3610)_{10}$

(g) $(0010111)_2 = (?)_{16}$ **Ans.** $(17)_{16}$

(h) $(6FA2)_{16} = (?)_2$ **Ans.** $(0110111110100010)_{16}$

(i) $(101111)_2 = (?)_8$ **Ans.** $(57)_8$

(j) $(532)8 = (?)_2$ **Ans.** $(101011010)_2$

(k) $(C45D)_{16} = (?)_8$ **Ans.** $(142135)_8$

(l) $(165)_8 = (?)_{16}$ **Ans.** $(75)_{16}$

Computational Thinking And Programming -1

2.1 INTRODUCTION TO PROBLEM-SOLVING

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 the 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

- Analysing 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. 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 → Continuously divide A with integers ranging from 2 to A-1

Step 3 → If a is divisible by any value from 2 to A-1, it is not prime

Step 4 → Else it 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

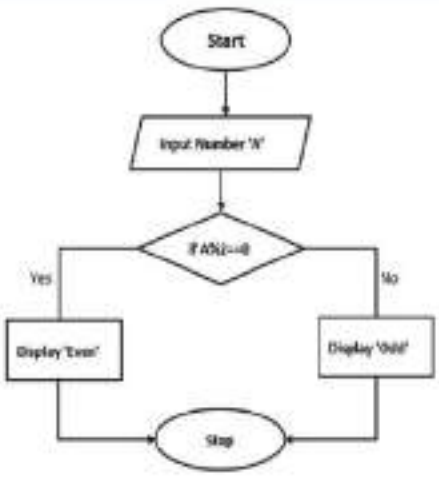
There are two common methods of representing an algorithm —flowchart and pseudocode.

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

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
START Step 1 → Take an integer number A as input Step 2 → Divide A by 2, and store the remainder as r Step 3 → If r is zero, Display 'Even' Step 4 → Else Display 'Odd' STOP	

Pseudocode

- Pseudocode is a way of representing an algorithm.
- 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:

- Easily convertible to a Programming Language
- 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.

2.2 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 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.

Display Output

print() function is used to display any message (or output) 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 previous 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.

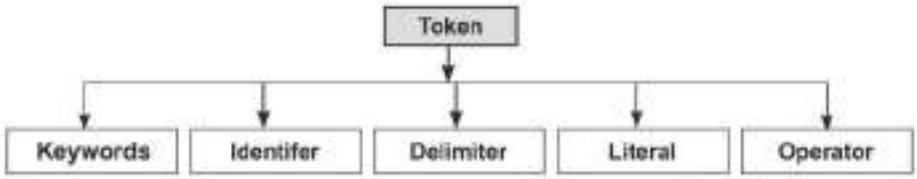
Character Set

A character set is a collection of valid characters that can be recognized by a language. Python Language recognizes 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 (which is already known to the interpreter). These words cannot 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 (which is given by Programmer) 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 identifiers.
4. First character cannot be a digit.

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.

Punctuators

Punctuators are symbols used to structure code and define syntax.

Example

- Parentheses ()
- Brackets []
- Curly braces {}
- Comma ,
- Colon :

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 (Non-Executable Commands/Instruction).

'#' is used to indicate that a Python statement is a single-line comment.

Variable

Variables refer to an object (data items like int, float, list, dictionary etc) stored in the memory. Value referred by 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.

2.3 KNOWLEDGE OF DATA TYPES

Data Type

Data type represents the type of data of 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.

Immutable Data Object
Integer
Float
Boolean
String
Tuple

Mutable Data Object
List
Dictionary
Set

Numeric Data Types (Number)

Python has three numeric data types:

- Integer
- Float
- Complex

Integer

- Numbers with No Fractional Part
- Can be Positive or Negative
- 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.

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.
- Elements of a list are 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 elements of a Tuple cannot be modified after creation.
- Elements of tuple are 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
elements
```

Dictionary:

- Dictionary in Python stores items in the form of key-value pairs
- Syntax : dict_variable = {key1:value1, key2:value2, ..., keyn:valuen}
- 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, we have to specify the key in the 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).

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.

2.4 Operators

Operators: Operators can perform operation on 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.

Operand: It is a variable or a value on which we perform the operation.

Expression

An expression is combination of operators, operands 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 like addition, subtraction, multiplication, division, floor division, exponent and modulus. (+,-,*,/,//,**,%)

Operator	Description	Example
+ (Addition)	Adds values on either side of the operator.	a=10 b=20 c=a+b print(c) will produce 30 as result
- (Subtraction)	Subtracts right side operand from left side operand.	a=10 b=20 c=b-a print(c) will produce 10 as result
* (Multiplication)	Multiplies values on either side of the operator	a=10 b=20 c=a*b print(c) will produce 200 as result
/ (Division)	Divides left side operand by right side operand	a=25 b=2 c=a/b print(c) will produce 12.5 as result

% (Modulus)	Divides left side operand by right side operand and returns the remainder	a=25 b=2 c=a%b print(c) will produce 1 as result
** (Exponent)	Performs exponential (power) calculation on operators	a=25 b=2 c=a**b print(c) will produce 625 as result
// (Floor Division)	Performs division, but digits after the decimal point in the quotient are removed.	a=25 b=2 c=a//b print(c) will produce 12 as result

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 are used to determine truth value based on multiple conditions. 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 and y
or	Logical OR	x or y
not	Logical NOT	not x

Assignment operators:

Variables are assigned values using assignment operators.

Assignment Operators		
Symbol	**Name**	**Syntax**
=	Assignment	x=y+z
+=	Addition assignment	x+=y (x=x+y)
-=	Subtraction assignment	x-=y (x=x-y)
*=	Multiplication assignment	x*=y (x=x*y)
/=	Division assignment	x/=y (x=x/y)

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 refer to 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.

2.5 EXPRESSIONS, STATEMENT, TYPE CONVERSION, AND INPUT/OUTPUT

Operator Precedence

Operator precedence determines the priority of operators in an expression.

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 Operators	Left-to-Right
in, not in, is, is not	Membership, Identity Operator	Left-to-Right
not	Logical NOT	Right-to-Left
and	Logical AND	Left-to-Right

or	Logical OR	Left-to-Right

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
```

Accepting Data as Input from the Console and Displaying Output in Python

We can accept data from the console using the input() function and display output using the print() function.

Accepting Data as Input

The input() function is used to take input from the user. It returns user input in string format.

Example

```
user_input = input("Enter something: ")
```

Displaying Output

The print() function is used to display output on the console. It can take multiple arguments, separated by commas, and prints them with a space in between.

Example

```
print("Hello, World!")
```

2.6 ERRORS

Error

An error is a problem that occurs in a program. Error sometimes halt the program execution, or produce unexpected results, or 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)
```

2.7 FLOW OF CONTROL

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 or equal to 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
```

2.8 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.

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.

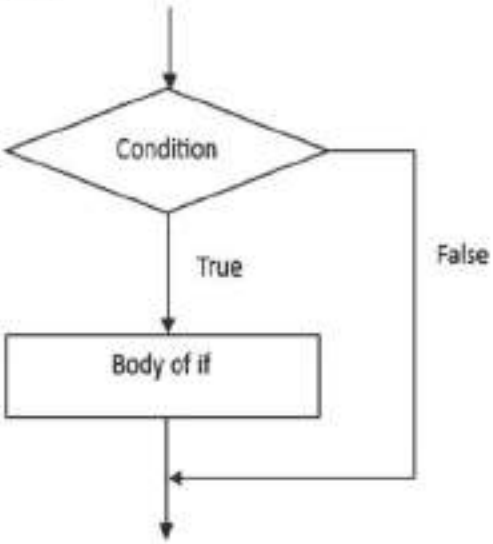
if statement

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



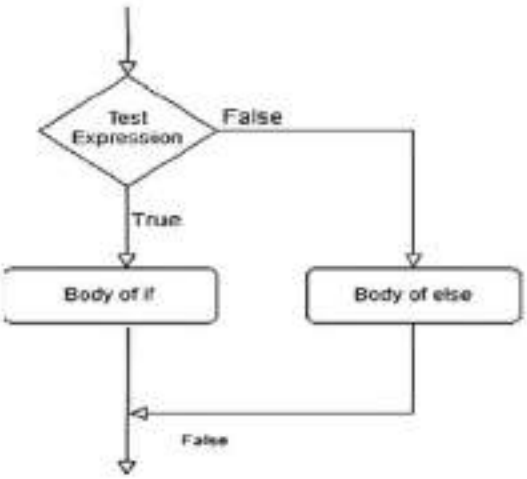
Example:

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

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

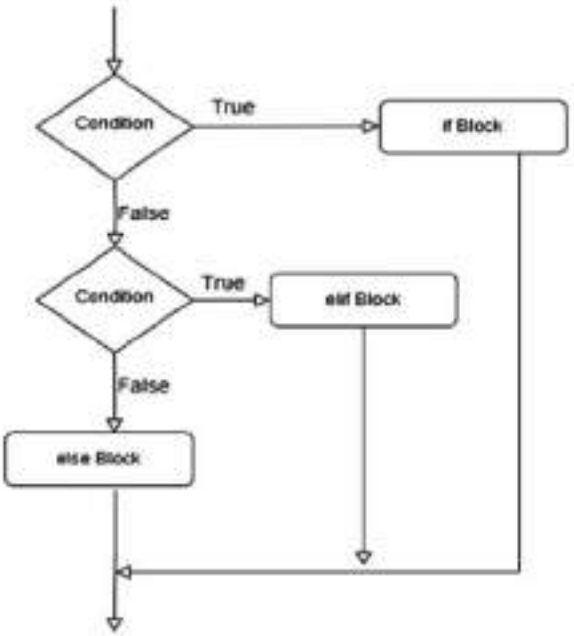
if-elif-else statement is used to check multiple conditions.

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



2.9 ITERATIVE STATEMENT

Iterative Statement

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

while Loop

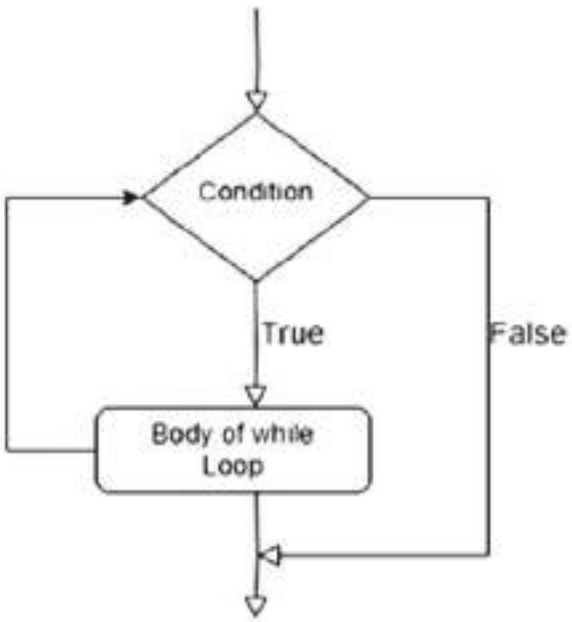
The while loop repeatedly executes a block of code as long as the specified condition is true.

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.It will create a sequence of values that starts from a specified start value, goes up to (but does not include) a specified stop value, and increments or decrements by a specified step value.

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

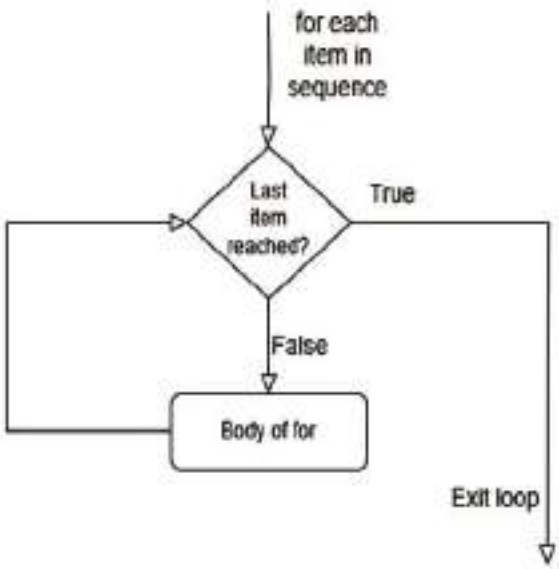
for Loop

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

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

Example

```
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")
else:
    print("Number should be greater than 1")
```

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 loop 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
*
**
***
****
*****

MID- TERM PRACTICE PAPER

Note: The sample paper provided below is based on the exam pattern specified by CBSE for Class XII Computer Science sample paper for session 2023-24. For the paper pattern of the 2024-25 session, please refer to the latest CBSE sample paper (for session 2024-25 session).

Class: XI

Mid Term Examination

Computer Science (083)

Sample Question Paper (Theory)

Time allowed: 3 Hours Maximum Marks: 70

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.

Q No. Section-A (18 Marks) Marks

1. Which statement about computer memory is correct? 1

   (a) RAM is volatile (b) ROM is volatile

   (c) Secondary Memory is volatile (d) RAM always retains its data

2. How much data does one terabyte (TB) represent? 1

   (a) 210 byte (b) 210 MB

   (c) 210 KB (d) 210 GB

3. Convert the decimal number $(345)_{10}$ to Octal: 1

(a) 331 (b) 431

(c) 531 (d) 631

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 ISCII 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. The return type of input() function is 1

(a) list (b) integer

(c) string (d) float

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 1

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. 2

OR

Expand the following (i.e. write Full Form of):

1. ALU
2. CPU

20. Write any two features of Python Programming language. 2

21. Explain the following with a suitable example of each 1+1=2

(a) Primary Memory (b) Secondary Memory

OR

Why is RAM called a volatile memory?

22. Write a Python program that accepts distance in distance in meters and converts it into centimetres. 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." 2

OR

What do you mean by error in python? Write down the name of any two types of error.

24. Define Data-Type and mention any 2 examples. 2

OR

Explain Implicit type conversion & Explicit type Conversion

25. Differentiate between Mutable and Immutable data objects in Python. Give one example of each. 2

Q No.	Section-C (15 Marks)	Marks

26. Explain the following 3

(i) Device Driver with any one example

(ii) Assembler

(iii) Output Devices with any one example

27. Perform the following conversions 3

(a) Convert $(77)_{10}$ to octal.

(b) Convert $(232)_8$ to binary.

(c) Convert $(74F)_{16}$ to decimal.

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. 3

OR

Write a python program that takes a number as input and prints even or odd accordingly.

29. Write a program in python that accepts the radius of a circle and prints its area. 3

30. Write the following Python code after removing all syntax error(s). Also underline all corrections made.

```
a;b=7,7
a+b=c
If c=> 14:
      print("More than double')
elif:
      Print("less than double")
```

3

Q No. **Section-D (8 Marks)** **Marks**

31. Define Encoding Scheme and explain the following schemes.

(a) ASCII

(b) ISCII

(c) Unicode 4

32. Consider the following python statement

n1, n2, n3 = 4, 3, 2

Consider the above variables (and values) for answering the questions given below.

(a) Write the output of
`x=2`
`x+=n2+n3`
`print(x)`
	(b) Write the output of
	`x=n1**(n2+n3)`
	`print(x)`

(c) Write the output of
`x=n1//float(10)`
`print(x)`
	(d) Write the output of
	`x='5'+'5'`
	`print(x)`

Q No. **Section-E (15 Marks)** **Marks**

33. Write a Python program to take sides of a triangle as input and print its area using Heron's Formula. (Hint : Area = Square root of(s(s - a)(s - b)(s - c))) 5

34. Write a program in Python that takes a 3 digit number as an input from user and checks and prints whether that number is an Armstrong number or not.

Hint: A three digit number is Armstrong if the number is equal to sum of cubes of its digits.

OR

Write a Python program to take temperatures in Fahrenheit as input and convert it to Celsius. (Hint C = 5/9(F-32)) 5

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.

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

2.11 STRINGS

Functions

These ae the built-in functions of Python. Functions are independent blocks of code that can be called from anywhere.

S. No.	Name of Function	Description
1.	len()	Returns the length of an object/Return the number of elements in the List

Methods of String

Methods are tied to objects and need an object to be invoked.

S. No.	Name of Method	Description
1.	capitalize()	converts the first character of a string to capital (uppercase) letter and rest in lowercase.
2.	title()	converts the first letter of every word in the string in uppercase and rest in lowercase
3.	lower()	Returns the string with all uppercase letters converted to lowercase
4.	upper()	Returns the string with all lowercase letters converted to uppercase

5.	count()	Returns number of times a substring occurs in the given string
6.	find()	Returns the index of the first occurrence of substring in the given string. If the substring is not found, it returns -1
7.	index()	Same as find() but raises an exception if the substring is not present in the given string
8.	isalnum()	Returns True if all characters in the string are alphanumeric (either alphabets or numbers). If not, it returns False
9.	isalpha()	Returns True if all characters in the string are alphabets, Otherwise, It returns False
10.	isdigit()	returns True if all characters in the string are digits, Otherwise, It returns False
11.	isspace()	Returns True if has characters and all of them are white spaces (blank, tab, newline)
12.	islower()	returns True if the string has letters and all of them are in lower case and otherwise False.
13.	isupper()	returns True if the string has letters and all of them are in upper case and otherwise False.
14.	strip()	Returns the string after removing the whitespaces both on the left and the right of the string
15.	lstrip()	Returns the string after removing the whitespaces only on the left of the string
16.	rstrip()	Returns the string after removing the whitespaces only on the right of the string
17.	replace (oldstr, newstr)	used to replace a particular substring in a string with another substring
18.	partition()	The partition() function is used to split a string into three parts based on a specified separator
19.	split()	Returns a list of words delimited by the specified substring. If no delimiter is given then words are separated by space.
20.	startswith()	Used to check whether a given string starts with a particular substring.
21.	endswith()	Used to check whether a given string ends with a particular substring

22.	join()	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.

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
- **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
  - The index can also be an expression including variables and operators but the expression must evaluate to an integer

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

Example

```
>>> str1 = 'Python'
>>> str1[0]
'P'
```

```
>>> str1 = 'Python'
>>> str1[2+3]
'n'
```

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

Case-1 print(str1[:3]) **Output** PYT	**Case-2** print(str1[1:4]) **Output** YTH	**Case-3** print(str1[0:5:2]) **Output** PTO	**Case-4** print(str1[-5:-1]) **Output** YTHO
Case-5 print(str1[-1:-4]) **Output** ''	**Case-6** print(str1[:-5:-1]) **Output** NOHT	**Case-7** print(str1[:-5]) **Output** 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 left to right or right to left.

1. **Getting characters of a String using for Loop**

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

2. **Getting characters of a String using index value, 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 : Adding two strings (Both operand should be string)

Operator : +

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

```
ch1='1'
ch2='1'
ch3=ch1+ch2
print(ch3)
output : '11'
```

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. It returns True or False.

Example

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

>>>"Prog" in str1	>>>"ming in" in str1	>>>"Pyth " in str1	>>>"Pyth " not in str1
True	True	False	True

String Methods/Functions

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

len()

Returns the length of the given string

```
>>>str1='Hello World!'
>>>len(str1)
12
```

capitalize()

converts the first character of a string to capital (uppercase) letter and rest in lowercase.

```
str1="python Programming for 11th"
str1.capitalize()
'Python programming for 11th'
```

```
>>>str1="python programming for 11th"
>>>str1.index('r')
8
>>>str1.index('u')
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
ValueError: substring not found
```

isalnum()

The isalnum() method returns True if all characters in the string are alphanumeric (either alphabets or numbers). If not, it returns False.

```
>>>str1='HelloWorld'
>>>str1.isalnum()
True
>>>str1='HelloWorld2'
>>>str1.isalnum()
True
```

isalpha()

Returns True if all characters in the string are alphabets, Otherwise, It returns False

```
>>>'Python'.isalpha()
True
>>>'Python 123'.isalpha()
False
```

isdigit()

returns True if all characters in the string are digits, Otherwise, It returns False

```
>>>'1234'.isdigit()
True
>>>'123 567'.isdigit()
False
```

isspace()

Returns True if has characters and all of them are white spaces (blank, tab, newline)

```
>>>str1.isspace()
>>>str1=' \n \t'
True
>>>str1='Hello \n'
>>>str1.isspace()
False
```

islower()

returns True if the string has letters and all of them are in lower case and otherwise False.

```
>>>str1='hello world!'
>>>str1.islower()
True
>>>str1='hello 1234'
>>>str1.islower()
True
>>>str1='hello ??'
>>>str1.islower()
True
```

isupper()

Returns True if the string has letters and all of them are in upper case and otherwise False

```
>>>str1='HELLO WORLD!'
>>>str1.isupper()
True
>>>str1='HELLO 1234'
>>>str1.isupper()
True
>>>str1='HELLO ??'
>>>str1.isupper()
True
```

strip()

Returns the string after removing the whitespaces both on the left and the right of the string

```
>>>str1='   Hello World!   '
>>>str1.strip()
'Hello World!'
```

lstrip()

Returns the string after removing the whitespaces only on the left of the string

```
>>>str1='   Hello World!'
>>>str1.lstrip()
'Hello World!'
```

rstrip()

Returns the string after removing the whitespaces only on the right of the string

```
>>>str1='   Hello World!    '
>>>str1.rstrip()
'   Hello World!'
```

replace(oldstr, newstr)

used to replace a particular substring in a string with another substring

```
>>>str1='Hello World!'
>>>str1.replace('o','*')
'Hell* W*rld!'
```

partition()

The partition() function is used to split a string into three parts based on a specified separator.

```
>>>str1='India is a Great Country'
>>>str1.partition('is')
('India ','is',' a Great Country')
>>>str1.partition('are')
('India is a Great Country',' ',' ')
```

split()

Returns a list of words delimited by the specified substring. If no delimiter is given then words are separated by space.

```
>>>str1='India is a Great Country'
>>>str1.split()
['India','is','a','Great','Country']
>>>str1='India is a Great Country'
>>>str1.split('a')
['Indi',' is ',' Gre','t Country']
```

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
```

2.12 LIST

Functions

These are the built-in functions of Python. Functions are independent blocks of code that can be called from anywhere.

S. No.	Name of Function	Description
1.	list()	Convert an object into list
2.	len()	Returns the length of an object/Return the number of elements in the List
3.	max()	Returns the largest element from the list/ iterable
4.	min()	Returns the smallest element from the list/ iterable
5.	sum()	Returns the sum of all elements of an iterator/list
6.	sorted()	Returns a sorted list

Methods of Python List

Methods are tied to objects and need an object to be invoked.

S. No.	Name of Method	Description
1.	append()	Adds an element at the end of the list
2.	extend()	Add the elements of a list (or any iterable), to the end of the current list
3.	insert()	Adds an element at the specified position/index
4.	count()	Returns the number of elements with the specified value
5.	index()	Returns the index of the first occurrence with the specified value
6.	remove()	Removes the first occurrence with the specified value
7.	pop()	Removes the element at the specified position
8.	reverse()	Reverses the order of the list
9.	sort()	Sorts the list

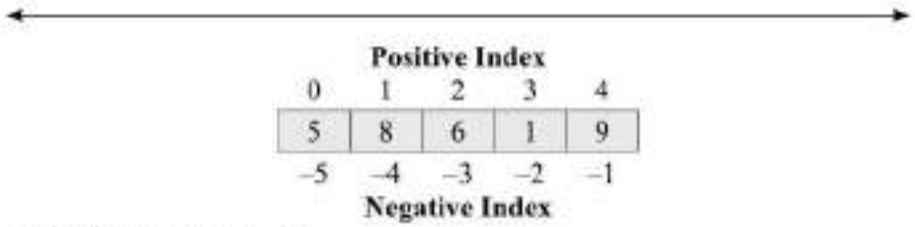
List:

- 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 of data.

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

Length = 5 (Number of elements)



What is the 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, list is 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 into an iterable type.
- Indexing can be used as positive indexing and negative indexing.
- Positive Indexes always 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:

```
+ve indexing
    0 1   2  3  4
L= [5, 8, 6, 1, 9]
   -5 -4  -3 -2 -1
-ve indexing
```

Access the element 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 give IndexError

```
Syntax:
Object_name[start : stop : interval]
```

- Here start, stop and interval all have default values.
- Default value of start is 0 (if interval is +ive) and -1 if interval is –ive
- Default value of interval is 1
- In slicing start is included and stop is excluded.
- If interval is +ive then slicing is performed from left to right.
- Slicing will follow either forward or backward traversal.

```
+ve indexing
     0  1  2  3  4
L = [5, 8, 6, 1, 9]
    -5 -4 -3 -2 -1
-ve indexing
```

```
>>> L[1:4:1]
[8, 6, 1]
```

If we give an index out of range then it is consider 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[: :]
[5, 8, 6, 1, 9]
```

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

```
>>> L[4:2]
[]
>>> L[4:4]
[]
>>> L[4:2:1]
[]
```

- If the interval is negative then slicing is performed from right to left. For Ex:

```
+ve indexing
     0     1     2     3     4
L = [5,    8,    6,    1,    9]
    -5    -4    -3    -2    -1
-ve indexing
```

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

```
>>>L[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

```
>>>L[5:0:-2]
[9, 6]
>>>L[-1:-4:-1]
[9, 1, 6]
>>>L[-1:-4:-1]
[9, 1, 6]
```

Access the list in reverse order

```
>>>L[: : -1]
[9, 1, 6, 8, 5]
>>> 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']
```

Using repetition (*) operator

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

Repetition Operator

- The repetition operator enables the list elements to be repeated multiple times.
- 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 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():

- len() is a function of list.
- 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 object into list.

For Ex:

```
>>> S="Python"
>>> L=list(S)
>>> L
['P', 'y', 't', 'h', 'o', 'n']
In the above example, a string S in converted into the List.
```

3: append()

- append() is a method of list, which is used to append 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 list.
- extend() iterates over its argument and adding 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 we extend a list with a string, it'll append each character as we 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 elements 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 that 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 first occurrence of the element.

Syntax: list_name.index(element, start, end)

element – The element whose first occurrence index will be returned.

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

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

```
>>> 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 does 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() will remove and return last element of the list, otherwise it removes and returns 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() 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 finds the minimum element from the list.

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

14: max()

max() function finds the maximum element from the list.

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

15: sum()

sum() 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.

```
      0            1                2          3
L = [1,      [2,    3,     4,],   'Python',   3.5]
             0      1      2      0 1 2 3 4 5
```

```
>>> 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:

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
```

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
```

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
```

4. Write a program in Python to count the frequency of the 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
```

2.13 TUPLE

Functions

These are the built-in functions of Python. Functions are independent blocks of code that can be called from anywhere.

S. No.	Name of Function	Description
1.	tuple()	Convert an object into tuple
2.	len()	Returns the length of an object/Returns the number of elements in the tuple
3.	max()	Returns the largest element from the tuple/ iterable
4.	min()	Returns the smallest element from the tuple/ iterable
5.	sum()	Returns the sum of all elements of an iterator/tuple
6.	sorted()	Returns a **sorted list**

Methods of Python Tuple

Methods are tied to objects and need an object to be invoked.

S. No.	Name of Method	Description
1.	count()	Returns the number of elements with the specified value
2.	index()	Returns the index of the first occurrence with the specified value

Tuple:

- Tuple is a sequence data type.
- Tuple is a heterogeneous collection of data.
- Tuple is an immutable data type i.e. it cannot be change after creation.
- 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 single element. Comma must be use after
the element.
>>> T= (5,)
>>> T
(5,)
>>> type(T)
<class 'tuple'>
#Without parenthesis comma separated values also treated as
tuple.
>>> T=5,
>>> T
(5,)
#Without comma, single element is not tuple.
>>> T = (5)
>>> T
5
>>> type(T)
<class 'int'>
#Following is string, not tuple
>>> T= ('Hello')
```

```
>>> T
'Hello'
>>> type(T)
<class 'str'>
>>> T= (5, 9.8,'Hello', 9)
>>> T
(5, 9.8, 'Hello', 9)
#Create a tuple without parenthesis.
>>> T=1, 2, 3, 4
 >>> T
(1, 2, 3, 4)
>>> type(T)
<class 'tuple'>
```

Tuple Indexing

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

```
>>> T= (5, 8, 6, 1, 9)
>>> T
(5, 8, 6, 1, 9)
>>> T[2]
6
>>> T[-3]
6
>>> T[6]
IndexError: tuple index out of range
```

Tuple Operations

1. Concatenation

Using + operator

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

2. Repetition

- The repetition operator enables the tuple elements to be repeated multiple times
- 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 element 1 and 2, The repetition operator makes 3 copies of this tuple and joins them all together into a single tuple.

3. 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 returns True if a element with the specified value is present in the tuple
2. **not in operator:** It returns True if a 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
```

4. 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:

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

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

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

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

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():

- 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 in converted into the tuple.
```

3: count()

count() is a function used count() the number of occurrence 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 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 returns the **smallest element** from the tuple.

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

7: max()

It returns the largest element from the tuple.

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

8: sum()

It returns the sum of the all elements of the tuple.

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

9: 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 nested inside the 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 list, even
it is nested inside the tuple.
>>> T[1][2]=7
>>> T
(5, [8, 6, 7], 3, 9)
```

Suggested Programs:

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
```

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
```

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
```

4: Write a program in Python to count the frequency of the 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
```

2.14 DICTIONARY

Functions

S. No.	Name of Function	Description
1.	dict()	It is used to create a dictionary
2.	len()	It returns the number of keys in the list.
3.	max()	It returns the largest key from the dictionary.
4.	min()	returns the smallest key from the dictionary
5.	sorted()	Return the keys of dictionary in sorted order
6.	del	del is a keyword. del is used to delete an element from the dict. and also be used to delete the complete dict.

Methods of Python Dictionary

S. No.	Name of Method	Description
1.	get()	It returns the value of the specified key.
2.	keys()	It returns the keys of the dictionary, as a list

3.	values()	It returns the values of the dictionary, as a list.
4.	items()	It returns the key-value pairs of the dictionary, as tuples in a list
5.	update()	It inserts specified elements to the dictionary or merge two dictionaries.
6.	clear()	It is used to delete all the elements
7.	pop()	It is used to delete a specific element from the dictionary and also return the value of the deleted element.
8.	popitem()	It is used to delete the last element from the dictionary.
9.	setdefault()	It returns the value of the specified key. If the key does not exist in the dictionary it inserts the key into dictionary.
10.	fromkeys()	It creates a dictionary with multiple keys which have the same value.
11.	copy()	It returns a copy of the specified dictionary.

Introduction to Dictionary:

- Dictionary is a mapping data type.
- Dictionary is a mutable data type i.e. it can be change after creation.
- Dictionary is denoted by curly braces i.e. {}
- Dictionaries are used to store data values in key : value pairs
- Keys of dictionary are unique and immutable.

```
>>> 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 key is duplicate at the time of dictionary creation then the
value assign at the last will be store 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 of a dictionary means that an element can be changed or deleted from 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

- We 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.

```
>>> 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.

```
>>> D = {1:10, 2:20, 3:30}
>>> D.values()
dict_items([(1, 10), (2, 20), (3, 30)])
#Print the 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 merges 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 already exists in the dictionary then the value of the existing key will be updated:

```
>>> 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 key name.

```
>>>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

```
>>>delD
>>>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 basis 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:

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}
```

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: 30000
Do you want to add more employee details(y/n): y
Enter employee Name: Shiva
Enter employee Salary: 35000
Do you want to add more employee details(y/n): n
{'Amit': 20000, 'Sachin': 25000, 'Saanvi': 30000, 'Shiva':
35000}
```

2.15 MODULES IN PYTHON

Functions/Methods in Modules

S. No.	Name of Module	Name of Functions/Methods/Attributes
1.	math module	sqrt(), ceil(), floor(), pow(), fabs(), sin(), cos(), tan() Attributes of math module: pi, e
2.	random module	random(), randint(), randrange()
3.	statistics module	mean(), median(), mode()

Introduction to Modules:

- A Python file with a .py extension 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.

   Syntax: import <module_name>

   For example: **import math**

   To access multiple modules, we can use following syntax

   Syntax: import module1, module2, ……, moduleN

   For example: import math, random

   To access particular methods from any module, use following syntax

   Syntax: <module_name>.<function_name>

   For example: random.randint(3,6)

2. **from statement**

   'from' statement is also used to import specific attributes or objects from a module. Imported methods will be directly used in program without module name.

   Syntax: from <module_name> import <function_name>

   OR

   Syntax: from <module_name> import function1, function2, ……..

   For example:

```
>>> from math import sqrt, floor
>>> sqrt(6)
2.449489742783178
```

3. **import ***

   import * statement used to import all objects from a module and access without the module name.

   from <module_name> import *

   For example:

   from math import*

   Module aliasing:

   You can create an alias (nickname) of a module using 'as' keyword. Methods of

modules will be accessed using alias name.

import <module_name> as <alias_name>

For example:

```
>>> import random as r
>>> r.randint(5,8)
>>> 6
```

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)
7.071067811865475
```

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 x^y

```
>>> 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 (π/180)

Where **π = 22/7 or 3.14**

Convert 30^0 in radians

30^0 in radians = Angle in degree x (π/180)

= 30 * 3.14 /180

=0.524 (approx)

Convert 60^0 in radians

60^0 in radians = Angle in degree x (π/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
```

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.

1. **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()
0.9578591158487608
>>> r.random()
0.45295811857945245
>>> r.random()
0.11874921693577878
```

2. **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)
6
>>> r.randint(4,9)
8
>>> r.randint(4,9)
7
```

3. **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)
7
#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)
5
```

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

1. **mean()**

   Syntax: **statistics.mean(data)**

   The mean() method returns the arithmetic mean (average) of the given data set.

2. **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.

3. **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
```

PRACTICE QUESTIONS (SOLVED)

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

Ans. (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

Ans. (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

Ans.(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

Ans.(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

Ans. (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

Ans. (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()

Ans. (d) len()

8. What is the output of the following Python code?

```
x = 5
y = 2
result = x/y
print(result)
```

(a) 7 (b) 2.5

(c) 2 (d) Error

Ans. (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

Ans. (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

Ans. (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

Ans.(b) 123_variable

12. Which operator is used to check whether two values are equal in Python?

(a) == (b) !=

(c) <= (d) >

Ans.(a) ==

13. What is the result of the expression 5%2 in Python?

(a) 2 (b) 2.5

(c) 0.5 (d) 1

Ans.(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

Ans.(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) >

Ans.(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

Ans.(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

Ans. (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']

Ans. (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

Ans. (c) 9

20. Suppose L is [5, 3, 9, 4], what is min(L)?

(a) 5 (b) 3

(c) 9 (d) 4

Ans. (b) 3

21. Suppose L is [5, 3, 9, 4], what is sum(L)?

(a) 5 (b) 3

(c) 9 (d) 21

Ans. (d) 21

22. Suppose L is [2, 3, 8, 4, 5], What is L[-1]?

(a) 2 (b) 5

(c) 8 (d) 4

Ans. (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]

Ans. (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) {}

Ans. (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))

Ans. (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]

Ans. (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

Ans. (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

Ans. (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

Ans. (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

Ans. (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

Ans. (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

Ans. (d) 91

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]

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

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.

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

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

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

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

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

UNSOLVED QUESTIONS – 01 MARK

1. **Which of the following operator is used to check if two values are equal?**

   (a) == (b) !=

   (c) < (d) >

2. **Which of the following operator is used to get the remainder?**

   (a) / (b) %

   (c) * (d) -

3. **Which data type is used to store textual data?**

   (a) Integer (b) Float

   (c) String (d) Boolean

4. **What operator is used to assign a value to a variable?**

   (a) == (b) !=

   (c) = (d) +=

5. **What data type is used to store a sequence of elements that cannot be changed?**

   (a) List (b) Dictionary

   (c) String (d) Tuple

6. **Which data type is used for collections of key-value pairs?**

   (a) Tuple (b) Dictionary

   (c) List (d) String

7. **What will be the output for the following statement:**

```
print(4 + 2 - (2 * 4**3) + 21//11)
```

   (a) 121 (b) -121

   (c) 121.0 (d) -121.0

8. **Which of the following is not a Python keyword?**

(a) with (b) if

(c) elif (d) iterate

9. **What is the correct syntax to create a tuple in Python?**

(a) tup1 = [1,2,3] (b) tup1 = {1,2,3}

(c) tup1 = (1,2,3) (d) tup1 = <1,2,3>

10. **Which of the following is used to denote comments in Python?**

(a) // (b) <!-- -->

(c) /* (d) #

11. **What will be the output of the following code?**

```
x = 5
print(x == 5)
```

(a) True (b) False

(c) 5 (d) Error

12. **What is the output of print(2 ** 3)?**

(a) 6 (b) 8

(c) 9 (d) 5

13. **Which of the following is a valid variable name in Python?**

(a) 1_variable (b) variable_1

(c) variable-1 (d) variable.1

14. **Which operator is used for multiplication in Python?**

(a) / (b) *

(c) - (d) %

15. **What will be the output of the following code?**

```
x = 10
y = 20
print(x is y)
```

(a) True (b) False

(c) 10 (d) 20

16. **What is the output of print(type(3.14))?**

(a) <class 'int'> (b) <class 'float'>

(c) <class 'str'> (d) <class 'complex'>

17. Which of the following data types is immutable in Python?

(a) list (b) dictionary

(c) set (d) tuple

18. Which function is used to read input from the user in Python?

(a) input() (b) get_input()

(c) read() (d) readline()

19. What will be the output of the following code?

```
x = 5
y = 10
print(x + y)
```

(a) 5 (b) 10

(c) 15 (d) 20

20. Which of the following is not a valid Python data type?

(a) int (b) float

(c) char (d) str

21. What is the output of print(10 // 3)?

(a) 3.33 (b) 3

(c) 4 (d) 10

22. What statement is used to execute a block of code repeatedly as long as a condition is true?

(a) if (b) elif

(c) for (d) while

23. What keyword is used to exit a loop prematurely?

(a) continue (b) break

(c) exit (d) stop

24. Find the output of the following python code:

```
c=0
for i in range(5):
      if (i%2==0):
           c = c+2
      if (i%4==0):
           c+=3
print(c)
```

(a) 10 (b) 11

(c) 12 (d) 13

25. Which of the following is not a loop in Python?

(a) for (b) while

(c) do-while (d) none of the above

26. What is the output of print("Hello" + " World")?

(a) HelloWorld (b) Hello World

(c) Hello + World (d) Error

27. Which of the following is not a valid string method?

(a) upper() (b) lower()

(c) capitalize() (d) makeupper()

28. How can we convert a string to lowercase in Python?

(a) lower() (b) tolower()

(c) lowercase() (d) downcase()

29. What will be the output of the following code?

```
x = "Hello"
y = x.lower()
print(y)
```

(a) HELLO (b) hello

(c) Hello (d) Error

30. What is the output of the following code?

```
print("Hello"[1])
```

(a) H (b) e

(c) l (d) o

31. What will be the output of the following code?

```
print("Hello, World!".upper())
```

(a) hello, world! (b) Hello, World!

(c) HELLO, WORLD! (d) Error

32. Which of the following is not a valid dictionary method?

(a) keys() (b) values()

(c) items() (d) append()

33. What is the correct way to create a dictionary in Python?

(a) {1, 2, 3} (b) {1: 'a', 2: 'b'}

(c) [1, 2, 3] (d) (1, 2, 3)

34. Which method is used to get a list of all keys in a dictionary?

(a) keys() (b) values()

(c) items() (d) all_keys()

35. Which of the following is a valid dictionary method?

(a) add() (b) update()

(c) append() (d) insert()

UNSOLVED QUESTIONS – 02 MARKS

1. Write the two modes of working in Python. Explain the difference between these modes.
2. Differentiate between Mutable and Immutable data objects in Python. Give one example of each.
3. Write any two differences between List and Tuple.
4. Write a python program to find whether the given is even or odd.
5. What will be the output of the following code?

```
for i in range(1, 5):
    print(i * '*')
```

6. Write a program in python to enter distance in centimeter and convert it into meter.
7. Write a Python program that takes two numbers as input and performs all arithmetic operations on them (addition, subtraction, multiplication, division, and modulus).
8. Write a Python program that takes a string input from the user and prints it in uppercase.
9. Explain syntax error, logical error, and runtime error in Python.
10. Write a Python program that prints numbers from 1 to 10 using a while loop. Explain the role of indentation in your program.
11. What will be the output of the following code?

```
import math
print(math.sqrt(16))
print(math.pow(2, 3))
```

12. Write a Python program using for loop to display the multiplication table of a given number.
13. Write a Python program to create a tuple with 5 elements and print each element using for loop.
14. Write a Python program to create a dictionary of 3 students with their names as keys and their scores as values. Print each student's name and score.
15. Write a Python program that uses the math module to calculate the area of a circle given its radius.

Unsolved Questions – 03 Marks

1. Explain the difference between syntax errors and logical errors in Python.
2. Write a program to input 3 sides of a triangle and find out whether the triangle is Equilateral, Scalene or Isosceles triangle?
3. Write a program to find the factorial of a number.
4. Write a Python program that creates a dictionary 'phonebook' with names as keys and phone numbers as values. Take details as user input and add two entries to the dictionary and display their details.
5. Write a Python program that accepts a string from the user and:
   - Counts the total number of characters (including spaces).
   - Converts the string to uppercase.
   - Reverses the string.
6. Given a list of integers, write a function that:
   - Finds the sum of all even numbers in the list.
   - Determines the maximum value in the list.
   - Removes duplicates from the list.
7. Write a Python Script to print Fibonacci Series of first 15 elements?
8. Input a list of numbers and swap elements at the even location with the elements at the odd location.
9. Input a tuple of elements, search for a given element in the tuple.

UNSOLVED QUESTIONS – 04 MARKS

1. Write a Python program to create a list of integers. The program should find and print the sum of all values, maximum value, minimum value, and average of the numbers in the list.

2. Explain the concept of immutability in tuples. Write a Python program that takes a tuple of numbers as input and prints the numbers in reverse order.
3. Fill in the blanks to create a program in Python that takes a string as input and returns the number of consonants in the string.

```
text = __("Enter a string: ") #Statement-1
consonants = "bcdfghjklmnpqrstvwxyz"
count = __ #Statement-2
for char in text:
    if char.lower() in consonants:
        count += __ #Statement-3
print("Number of consonants:", __) #Statement-4
```

   A. Write the name of the function to be used in the blank space in Statement-1.
   B. Write the value to be used in the blank space in Statement-2.
   C. Write the value to be used in the blank space in Statement-3.
   D. Write the variable name to be used in the blank space in Statement-4.
4. Consider the code given below and answer the questions

```
import random
NUMBERS = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
end = random.randrange(3) + 4 #Statement-1
begin = random.randint(0, end - 2) #Statement-2
for i in range(begin, end):
    print(NUMBERS[i])
```

   A. What will be the minimum possible value of begin (Refer Statement-2)?
   B. What will be the maximum possible value of begin (Refer Statement-2)?
   C. What will be the minimum possible value of end (Refer Statement-1)?
   D. What will be the maximum possible value of end (Refer Statement-1)?

Unsolved Questions – 05 Marks

1. Write a Program to count frequency of each character in a word using dictionary.
2. Write a Program to perform linear search on a tuple of numbers.
3. Differentiate between syntax errors, logical errors, and runtime errors. Give an example of each type of error and explain how we can fix these errors in our code.
4. Write a Python program to determine the grade of a student based on their marks (using an if-elif-else structure). The program should take marks of five subjects

as input and print the corresponding grade letter (A, B, C, D, or F) based on the percentage.

5. Create a Python program that accepts a list of numbers as input. For each element, determine whether it is even or odd. Store the even numbers in a list named even_ele and the odd numbers in a list named odd_ele. At last, print the contents of both lists.

Assertion and Reason Type Questions

1. (a) Both A and R are true and R is the correct explanation for A
2. (a) Only A is true.
3. (a) Both A and R are true and R is the correct explanation for A
4. (c) A is True but R is False
5. (d) Both A and R are False
6. (d) A is False but R is True
7. (d) Both A and R are False

UNIT
3

Society, Law and Ethics

3.1 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 behaviour, Payment information

How are digital footprints created?

Digital footprints 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.

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, and 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

- 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)

Trademark

Trademark is a visual symbol, name, design, slogan, label, etc., that distinguishes the brand or commercial enterprise from other brands or commercial enterprises.

Example trademark of Gmail, McDonald's 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.

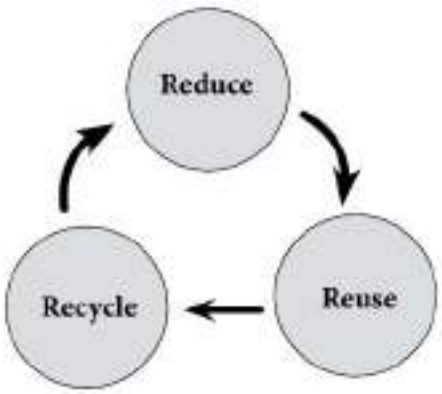
E-waste management cycle:

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

1. **Digital footprints are only created by social media activity, such as posting photos and comments.**

Ans. False.

2. **If you delete your information from the internet then your digital footprint is also deleted.**

Ans. False. Deleted information is difficult to trace but it may be accessible through backups or cached versions.

3. **Plagiarism is considered a violation of intellectual property rights, even if the copied material is not used for commercial purposes.**

Ans. True.

4. **The Creative Commons license allows creators to retain their copyright while permitting others to use their work under certain conditions.**

Ans. True.

5. **Viruses can spread through email attachments and infect computers when the attachment is opened.**

Ans. True.

6. **Trojans are self-replicating programs that can spread across a network without user intervention.**

Ans. False.

7. **Phishing is a form of cybercrime where attackers attempt to deceive individuals into disclosing sensitive information through fraudulent emails or websites.**

Ans. True.

8. **Ransomware is a type of malicious software that locks users out of their systems or encrypts their files until a ransom is paid.**

Ans. True.

Multiple Choice questions

1. **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

Ans.: (b)

2. 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

Ans.: (c)

3. 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

Ans.: (d)

6. 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

Ans.: (b)

7. 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.

Ans.: (a)

8. 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.

Ans.: (b)

9. 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.

Ans.: (c)

10. 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.

Ans. : (c)

11. 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

Ans. : (b)

12. What type of intellectual property right grants exclusive rights to inventors for their inventions?

(a) Copyright (b) Trademark

(c) Patent (d) Creative Commons

Ans.: (c)

13. 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.

Ans.: (b)

14. 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

Ans.: (d)

Very short answer type

1. What is a digital footprint?

Ans. A digital footprint is the trail of data and information left by an individual's online activities.

2. How is a digital footprint created?

Ans. Digital footprints are created through online interactions, such as social media posts, website visits, and online purchases.

3. What are Intellectual Property Rights (IPR)?

Ans. IPR refers to legal rights that protect creations of the mind, such as inventions, artistic works, and brand names.

4. What is the purpose of Creative Commons licenses?

Ans. Creative Commons licenses provide creators with a way to share their work under specific terms and conditions.

5. What is malware?

Ans. Malware is malicious software designed to harm computer systems or steal data.

6. How does a Trojan operate?

Ans. A Trojan disguises itself as legitimate software to trick users into installing it and then performs malicious actions.

7. How can users protect themselves from malware?

Ans. Users can install reputable antivirus software, keep systems and applications up to date, and avoid downloading files from unknown sources.

8. What is the Information Technology Act (IT Act)?

Ans. The Information Technology Act is a legal framework that addresses electronic transactions, cybersecurity, and digital data protection.

9. What are some objectives of the IT Act?

Ans. The objectives of the IT Act include providing legal recognition for electronic transactions, facilitating e-governance, and preventing cybercrimes.

10. How can technology help address gender disparities in education?

Ans. Technology can provide equal learning opportunities for all genders and promote access to education and professional opportunities.

Short Answer Type Question

1. How does an individual's digital footprint impact their online privacy and personal brand? Provide an example to support your answer.

Ans. 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.

2. Explain how digital footprints are created and expanded through online interactions.

Ans. 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.

3. **How can individuals and organizations ensure they are not violating intellectual property rights while using copyrighted materials for educational or commercial purposes?**

Ans. 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.

4. Discuss the importance of open-source software in fostering innovation and collaboration within the technology community.

Ans. Open-source software encourages sharing and collaboration, leading to faster development, higher quality, and a more extensive range of software solutions accessible to everyone.

5. **How can users differentiate between a legitimate software application and a potentially harmful trojan or malware?**

Ans. Users should only download software from reputable sources, check reviews, and verify the publisher's authenticity to avoid installing trojans or malware.

UNSOLVED QUESTIONS

Short Answer Questions:

1. What constitutes a digital footprint?
2. Identify the two main types of digital footprints.
3. What kind of information does a website collect to create a digital footprint?
4. Can you give an example of how an active digital footprint is created?
5. How is a passive digital footprint generated?
6. What steps can be taken to minimize a passive digital footprint?
7. Who is referred to as a netizen?
8. What are the rules called that govern respectful online behavior?
9. Provide an example of an ethical rule of net etiquettes.
10. Why is it important to respect privacy in the digital world?
11. What does the term 'cyberbullying' refer to?
12. How would you describe an internet troll?
13. What is the best way to deal with internet trolls?
14. What is meant by 'digital communication'?
15. Why should messages in digital communication be precise?
16. How can one ensure politeness in digital communication?

17. Why is credibility important in digital communication?
18. List two important social media etiquettes.
19. Why should you choose passwords carefully?

Long Answer Questions:

1. Explain the concept of digital footprints and distinguish between active and passive digital footprints.
2. Describe the types of information collected during online shopping and how it contributes to your digital footprint.
3. Discuss various ways to minimize a passive digital footprint.
4. What are net etiquettes and why are they important for a digital society?
5. Explain the significance of respecting privacy in both physical and digital worlds.
6. Describe how cyberbullying can impact individuals and how it can be prevented.
7. Define digital communication etiquettes and provide examples of good practices.
8. Discuss the role of social media etiquettes in maintaining a positive online presence.
9. Explain the concept of data protection and why it is crucial in the digital age.
10. What is Intellectual Property Right (IPR) and how does it protect creators?
11. Compare and contrast copyrights, patents, and trademarks.
12. Discuss the implications of plagiarism and copyright infringement in the digital world.
13. What is Free and Open Source Software (FOSS) and how does it benefit the software community?
14. Differentiate between freeware and proprietary software.
15. Explain the role of public licenses like Creative Commons (CC) and GNU GPL in the software industry.
16. Define cybercrime and provide examples of different types of cybercrimes.
17. Discuss the concept of hacking and its potential consequences.
18. What is eavesdropping and how can it be prevented in digital communication?
19. Describe phishing and ways to protect oneself from phishing attacks.
20. Explain ransomware and its impact on individuals and organizations.
21. Discuss the importance of cyber safety and measures to ensure safe browsing.
22. Explain identity theft and how to protect against it online.
23. What is malware and what are its different forms? Provide examples.
24. Describe the environmental impact of e-waste and methods for its management.
25. Explain the Information Technology Act and its key provisions.

Question Paper (Unsolved)

Note: The Question paper provided below is based on the exam pattern specified by CBSE for Class XII Computer Science sample paper for session 2023-24. For the paper pattern of the 2024-25 session, please refer to the latest CBSE sample paper (for session 2024-25 session).

Class: XI

Computer Science (083)

Question Paper (Theory)

Maximum Marks:70 **Time Allowed: 3 hours**

General Instructions:

1. *This question paper contains five sections, Section A to E.*
2. *All questions are compulsory.*
3. *Section A has 18 questions carrying 01 marks each.*
4. *Section B has 07 Very Short Answer type questions carrying 02 marks each.*
5. *Section C has 05 Short Answer type questions carrying 03 marks each.*
6. *Section D has 03 Long Answer type questions carrying 05 marks each.*
7. *Section E has 02 questions carrying 04 marks each.*
8. *All programming questions are to be answered using Python Language only.*

SECTION A

1. State True or False.

/= is a valid operator in python. (1)

2. 1 GB is equivalent to how many bytes? (1)

(a) 1024 (b) $(1024)^2$

(c) $(1024)^3$ (d) $(1024)^4$

3. Computer understands only _____ language. (1)

(b) English (b) Binary

(c) Unary (d) Ternary

4. In Boolean algebra, which law explains (X+Y)'=X'Y'? (1)

(a) Absorption Law (b) Distributive Law

(c) Idempotent Law (d) De Morgan's Law

5. Which of the following is a valid variable name? (1)

(a) 21Hello (b) Hello_21

(c) Hello 21 (d) Hello@

6. What will be the output of the following code: (1)

```
T=(1,2,3,4,5,6,7,8,9)
print(T[1:7:2])
```

(a) (2,4,6) (b) (2,4,6,8)

(c) (1,3,5) (d) (1,3,5,7)

7. What is the output of the following code? (1)

```
a=5
for i in range(a):
continue
a=a+i
print(a)
```

(a) 15 (b) 10

(c) 5 (d) 6

8. To use the mean() function, _____ module needs to be imported in a python program. (1)

(a) random (b) statistics

(c) math (d) pandas

9. Which function is used to add a single element at a particular position in a given list? (1)

(a) append() (b) extend()

(c) insert() (d) None of these

10. What will be the output of given python code? (1)

```
L = [1,2,3,(4,5),'India',[7,8,[9,10],11],13]
print(len(L))
```

(a) 13 (b) 7

(c) 12 (d) 6

11. Which of the following is an invalid statement?

(a) Z="Class"*3 (b) Z="Class"+3

(c) Z="Class"+"Students" (d) Z=len("Class")-3

12. Can a dictionary have duplicate keys?

(a) Yes (b) Only if values are also same

(c) No (d) Only if the keys are strings

13. Which of the following is an example of an output device? (1)

(a) Projector (b) Keyboard

(c) Mouse (d) Microphone

14. Which of the following gives the same result as (A')'? (1)

(a) A (b) A'

(c) 0 (d) 1

15. Unwanted emails such as promotional emails sent to a large number of recipients are known as _____? (1)

(a) Virus (b) Spyware

(c) Adware (d) Spam

16. Which of the following is NOT a way to leave digital footprints? (1)

(a) Social Media Activity (b) Online Shopping

(c) Online Chatting (d) Any offline activity

17 and 18 are ASSERTION (a) AND REASONING (R) based questions. Mark the correct choice as

(a) Both A and R are true and R is the correct explanation for A

(b) Both A and R are true but R is not the correct explanation for A

(c) A is True but R is False

(d) A is false but R is True

17. Assertion (a): Regularly updating software and applications are important for cyber safety. (1)

Reasoning (R): Security patches are often included in software updates which makes it difficult for hackers to breach the system.

18. Assertion (a): Python is considered as interpreted language. (1)

Reasoning (R): In python, source code is executed in one go by the python interpreter by converting high level language into machine understandable language i.e. binary.

SECTION B

19. What will be the output of the following python code: (2)

```
a={ 1: "One" , 2: "Two" , 3: "Three" , 4: "Four"}
for x in a:
print (x, end= "@" )
```

20. What will be the output of following expression: (2)

5+3%2-10//3

OR

What will be the output of following code:

A= "Thisismycountry"

print(A.partition("my"))

21. Sonam is a Python programmer who is working on a program to print odd numbers between 1 and 100 (including 1 and 100). She wrote a python program for the same using a while loop. But now she wants to rewrite the code using a for loop. So, Help her to write the code using a for loop. (Note: Output of code using for loop should be same as code using while loop) (2)

```
a=1
while(a<101):
if a%2!=0:
    print(a)
a=a+1
```

22. Differentiate between RAM and ROM. (2)

23. Differentiate between sort() and sorted() functions of List. (2)

24. Consider two dictionaries – (2)

A={25:19, 3: 50, 16:60 , 2:'delhi'} and B={25:11, 2:20, 3:30}.

What will be the content of B after executing the following command?

B. update(A)

OR

What does the following code print?

a='cat' not in 'hello'

if not a:

print("Earth")

else:

print("Mars")

25. Mahesh works as a clerk in a government office and makes the salary of government employees. Recently the government increased the DA (Dearness Allowance) by 4%. So Mahesh wants to recalculate the salary of all government employees by adding this additional 4% DA in salary. Help him with this calculation by making a python program. The program takes Basic salary as an input from the user and calculates and prints the net salary.

Note: Net salary = Basic Salary + DA + HRA

Here, DA (Dearness Allowance) = 46% of Basic Salary

HRA (House Rent Allowance) = 27% of Basic salary 2

SECTION C

26. (a) Perform the following conversion and fill in the blank: (1+2=3)

$(FAB)_{16} = ()_2$

(b) Explain System software and application software with the help of an example.

27. Mayank is a Python programmer who is working on a program to find out and print the maximum number among 3 numbers. The code given below has some syntactical errors. Rewrite the code after removing errors and underline the corrections made. (Consider all three input numbers are different) (3)

```
A=int(input("Enter first number"))
B= int(input("Enter second number")
C= input("Enter Third number")
if A>B and A<C:
   print(A,"is the largest number")
elseif B>A OR B>C:
   print(B,"is the largest number)
else
print(B, "is the greatest number")
```

28. Write a program in python to print first n numbers of Fibonacci series. Value of n should be given by the user.

Note: Fibonacci series is an integer sequence of 0, 1, 1, 2, 5, 8, 13, 21 ...

Here The first two terms are 0 and 1. All other terms are obtained by adding the preceding two terms.

OR

Write a python program to take two List of integers L1 and L2 as an input from the user. Merge L1 and L2 to make a new list L3. Now count and display even and odd numbers present in list L3.

Example: If L1=[1,2,4,6,9] and L2=[2,4,7,9,5,98] then merge L1 and L2 to make L3=[1,2,4,6,9,2,4,7,9,5,98]. Then the count of even numbers is 6 and the count of odd numbers is 5. 3

29. Consider the following program. (2+1)

```
import random
L= [10,12,15,17,21,30,70];
Y = random.randint(1,3)
Z= random.randrange(2,4)
for i in range(Y, Z+1):
print(L[i], end = "#")
```

(a) What possible output(s) can't be displayed on screen when the above program is executed?

(i) 12@15@317@ (ii) 10@12@15@

(iii) 15@ (iv) 12@15@

(b) Specify the maximum values that can be assigned to the Y and Z variable.

30. Mohit is a novel writer and writes fictional stories. He just finished his latest novel and asked a few of his friends and family members for proofreading and suggestions. Then he contacted a publisher topublish his novel. There he came to know that his story has already been published by some other writer. Now, Mohit is not sure about what legal actions he can take to prevent his Intellectual Property Rights. (1×3=3)

(a) What type of intellectual property rights, including copyrights, patents, and trademarks is suitable for Mohit's Novel.

(b) Explain the concept of plagiarism in the context of the copied Novel.

(c) Explain the concept of copyright infringement in the context of the copied Novel.

SECTION D

31. Write a python program to take a string S as an input from the user and perform the following task:

(i) Replace character 'a' with 'e' and 'A' with 'E' present in string S.

(ii) Count the number of non-character values present in string S.

Example: if user input string S as 'Aman126', then value of S after replacement of characters will be 'Emen126' and count=3 4

32. Explain following terms in detail (any 4): (4)

(a) Phishing (b) Adware

(c) Firewall (d) Cyber Trolling

(e) Spam

SECTION E

33. (a) Differentiate between break and continue statements with the help of an example.

(b) Differentiate between pop() function and del statement in List with the help of an example.

(c) Write any 2 rules of declaring a variable in python? 2+2+1=5

34. Write a program in python that takes a line (string)and an input from the user and prints the following: (5)

(a) Number of uppercase letters

(b) Number of lowercase letters

(c) Number of spaces

(d) Number of digits

(e) Convert lowercase letter into uppercase and vice versa and then print the string.

OR

(a) Write a program in python to print the following pattern using loop:

1

12

123

1234

12345

(b) Write a program in python that will take a three digit number as an input from the user and display the sum of all the three digits. (e.g. input number is 234 so output will be (2+3+4=9)

35. (a) What etiquettes one should follow while using social media? (2+2+1=5)

(b) Why is proper handling of E-Waste necessary?

(c) What is Ransomware?

QUESTION PAPER (UNSOLVED)

Note: The Question paper provided below is based on the exam pattern specified by CBSE for Class XII Computer Science sample paper for session 2023-24. For the latest paper pattern of the 2024-25 session, please refer to the latest CBSE sample paper (for session 2024-25 session).

Class: XI

Computer Science (083)

Question Paper (Theory)

Maximum Marks:70 **Time Allowed: 3 hours**

General Instructions:

- *Please check this question paper contains 35 questions*
- *This paper is divided into 5 sections, A, B, C, D and E.*
- *All questions are compulsory.*
- *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.*

Q. No.	Question	Marks

SECTION A

1. State True or False. (1)

"The output of AND gate is false only if both the inputs are true"

2. Which unit of the CPU performs all the logical and arithmetical operations? (1)

(a) Keyboard (b) CU

(c) ALU (d) Mouse

3. _____ converts program written in assembly language into binary language. (1)

(a) Interpreter (b) Compiler

(c) Converter (d) Assembler

4. Which of the following is not a valid variable name? (1)

(a) abc (b) abc 12

(c) abc12 (d) abc 12

5. How many times will the following loop print 'hello'? (1)

```
for a in range(1,20,3):
print('hello')
```

(a) 7 (b) 6

(c) 20 (d) 9

6. Aman wants to print his name 20 times. So, He writes a python code for the same. But he is not able to complete it. Help aman to complete the code by filling in the blank.

```
a=1
while(a<40):
print('Aman')
    a=a+_____
```

(a) 1 (b) 2

(c) 3 (d) 0.1

7. Consider the following python statements and then choose the correct output option from the given options: (1)

```
x=[1,2,3,4]
y=[5,6,7]
x.append(y)
print(x[4])
```

(a) 5 (b) 4

(c) [5,6,7] (d) [5,6]

8. Consider the following tuple T and choose the correct option to print 8 from the tuple T. (1)

T = (1,2,4,[5,6,8,10])

(a) T[5] (b) T[3][2]

(c) T[3:2] (d) T[3,2]

9. Which of the following is not a valid function in tuple? (1)

(a) len() (b) sorted()

(c) pop() (d) sum()

10. Which of the following statement will give error?

```
z=(1,2,3,4,5)          #Statement 1
z[1]=6                 #Statement 2
```

z=(6,7,8) #Statement 3

print(z) #Statement 4

(a) #Statement 1 (b) #Statement 2

(c) #Statement 3 (d) #Statement 4

11. What will be the output of following python code: (1)

```
A='a'
B='abc'
print(A in B)
```

(a) a (b) abc

(c) A in B (d) True

12. Consider the following code and find out how many time string 'Hello' will be printed? (1)

```
Z=1
while Z<10:
print('Hello')
```

(a) 1 (b) 10

(c) Infinite (d) Error

13. _____ is the smallest unit to represent a character or data item. (1)

(a) Byte (b) Bit

(c) KB (d) MB

14. _____ is a high speed memory that the CPU can access more quickly than it can access RAM. (1)

(a) Pendrive (b) DVD

(c) Cache (d) Hard Disk

15. Which of the following is an example of e-waste? (1)

(a) Torn paper (b) Empty water bottle

(c) Old phone (d) Old clothes

16. Which of the following activities is an example of cyberbullying? (1)

(a) Ramesh sent an email to his friend with a message "hello".

(b) Ramesh sent a message to his friend saying "Do not try to call or talk to me".

(c) Ramesh created an embarrassing picture of his friend and uploaded on a social networking site

(d) Ramesh uploaded an old photo of his friends with caption "Miss you all"

Assertion and Reason:

In the following questions, A statement of Assertion (a) is followed by a statement of Reason (R). Mark the correct choice as:

(a) Both A and R are true and R is correct explanation of A

(b) Both A and R are true but R is not correct explanation of A

(c) A is true but R is false

(d) A is false but R is true

17. **Assertion (a):** Antivirus software is required to be installed in our computer to prevent computer viruses and other online threats. (1)

Reason (R): Hacking and other online frauds are increasing day by day.

18. **Assertion (a):** Addition and deletion of elements are not allowed in tuple. (1)

Reason (R): Tuple is a mutable data structure.

SECTION B

19. What will be the output of the following python code: (2)

```
L1=[10,20,30]
L2=[50,60,70]
L1.append(L2)
L1.extend(L2)
print(L1)
```

20. What will be the correct order of operator precedence in following expression. Also, evaluate the expression and produce output.

```
5%3+4**2
```

OR

```
a,b,c=2,5,7
c,b=b+2,b-2
a=c+b
print(a,b,c,sep="@")
```

2

21. Rewrite the following code using while loop so that it will generate the same output. (2)

```
for i in range(1,10,2):
print(i*10,end=' ')
```

22. Draw a logic circuit for the following Boolean expression: (2)

A'B+(CD)'

23. What do you understand by mutable and immutable data types? Explain with the help of an example. (2)

24. What will be the output of the following code?

```
L=["Hello","there",[10,50,88],"Computers"]
L[-1]="CS"
print(L[1:10])
```

OR

What will be the output of the following code? (2)

```
D={1:2,3:4,5:6,7:8}
L=[1,2,3,4,5]
for i in D:
L.append(i*2)
print(L)
```

25. Write a program in python that accepts a tuple of integers T from the user and then prints maximum number among them. (2)

Example: if T=(1,4,3,6,8,5,2)

Then, Maximum = 8

SECTION C

26. (a) erform the following conversion and fill in the blank: (1+2=3)

$(755)_8 = ()_{16}$

(b) Differentiate between compiler and interpreter.

27. Rewrite the following code after removing syntax errors (Underline the changes) to print the even numbers greater than 5 from the given list of integers: (3)

```
a=[1,2,3,4,5,6,7,8,9,10]
for b in a
IF b%2=0 & b>5:
print(b)
else:
CONTINUE
```

28. Write a program in python that repeatedly asks the user to input Student's Name and Class and store it in a dictionary D whose keys are Student's Name and values

are Student's Class. Then display the dictionary.

OR

Write a program in python that accepts students' names in the form of a tuple as an input from the user and then displays only those students' names that start with letter 'S' or 's'. (3)

29. Consider the following python code and answer the questions: (3×1=3)

```
import random
import math
A=random.random()
B=random.randint(1,5)
C=math.ceil(5.2)
```

(a) What could be the minimum and maximum value that can be assigned to B.

(b) What value assigned to C.

(c) State True or False, "Value of A could be anything between 1 and 2".

30. Mahima is a high school student. She is an active user of social media platforms. She was in a hill station for her last vacation. She posted all photos, videos, and other vacation information online. One day, she noticed that all of her private photos and information has been used to make a fake account of her without her permission. Now, Mahima is concerned about her digital footprint and its potential consequences. (3)

(a) What is a digital footprint?

(b) What type of cybercrime Mahima is a victim of?

(c) As a friend of Mahima,give her any one adviceto better manage and protect her digital footprint to prevent such future incidents?

SECTION D

31. Write a program in python that takes a number as an input from user and checks and prints whether that number is an Armstrong number or not.

Note: Armstrong number is a number that is equal to the sum of cubes of its digits.

Example 1: If input number is 124 then 13+23+43= 73. As 124 ≠ 73, So, 124 is not an Armstrong number.

Example 2: If input number is 153 then 13+53+33= 153. As 153 = 153, So, 153 is an Armstrong number. 4

32. Explain following terms in detail (any 4): (4×1=4)

(a) Plagiarism (b) Open source software

(c) Hacking (d) Cyber Stalking

(e) IT Act

SECTION E

33. (a) What do you understand about an infinite loop? Give an example that runs infinite times. (2+3=5)

(b) Explain following string functions with the help of an example:

(i) len() (ii) isupper()

(iii) count()

34. (a) Write a program in python to take a number as an input from the user and prints its table

Example: if input number is 7, then the output should be

7 x 1 = 7

7 x 2 = 14

7 x 3 = 21

7 x 4 = 28

7 x 5 = 35

7 x 6 = 42

7 x 7 = 49

7 x 8 = 56

7 x 9 = 63

7 x 10 = 70

(b) Write a program in python that accepts a dictionary D as an input from the user and then store all the values (from key:value pair) of dictionary D in a list L and then print L.

Example: if D={1:'aman',2:5,'hill':6,9:99}

Then L = ['aman',5,6,99]

OR

(a) Write a program in python that takes a string as an input from the user and displays it if its length is greater than 5.

(b) Write a program in Python that takes a list of integer from the user and displays the number of even elements 2+3=5

35. (a) What is cybercrime? What role does the IT Act play in preventing cyber crimes? (3+2=5)

(b) What do you understand by eavesdropping?

APPENDIX

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.

Running Python

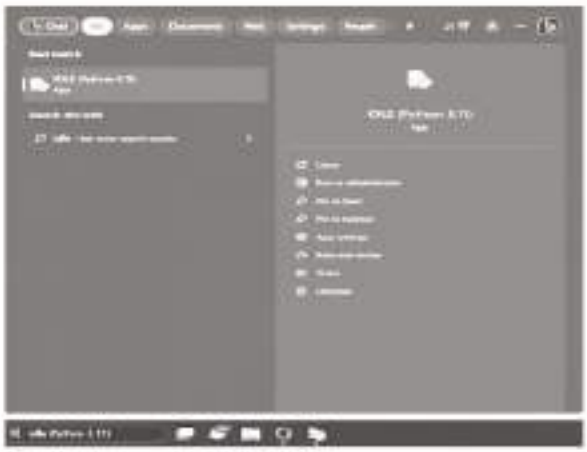
Using IDLE in Interactive Mode

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.

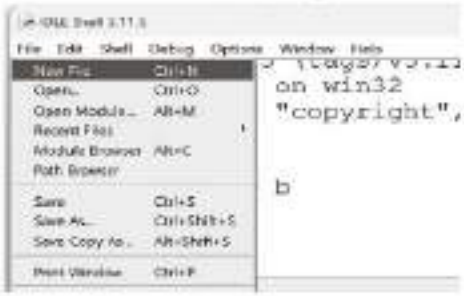
   >>> shows the prompt where Python statements can be written in the Interactive Mode.
3. The outputs will be immediately displayed after the entered statement.

Using Script Mode in IDLE

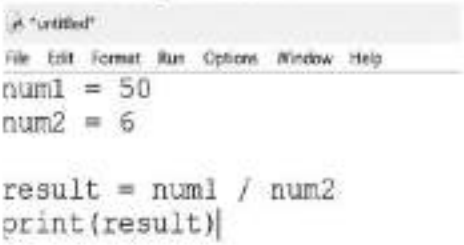
1. To start writing code in the Script Mode, start IDLE in Interactive Mode.



2. On the Menu Bar, choose File > New File, to open a new window.

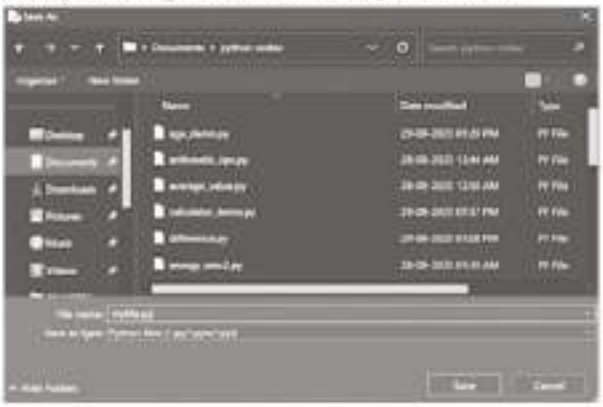


3. Start writing your code in the script window.

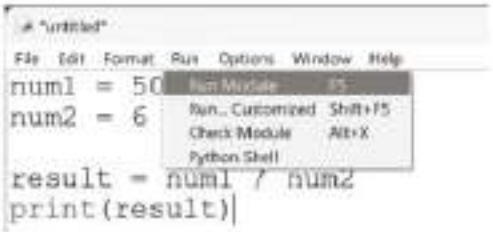


4. Save your Python Script by going to **File > Save**, and give any valid file name to your script with the **extension .py**

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.

```
>>>
======= RESTART: C:/Users/Diwakar/Documents/python
codes/myfile.py =======
8.333333333333334
>>>
```

8. To Open an existing Python script, choose File > Open and double click on the python file to Open it in IDLE Script Mode.

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.

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.

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.