कंप्यूटर विज्ञान
Computer Science

कक्षा / Class XI

2025-26

विद्यार्थी सहायक सामग्री
Student Support Material

केन्द्रीय विद्यालय संगठन~Kendriya Vidyalaya Sangathan

संदेश

विद्यालयी शिक्षा में शैक्षिक उत्कृष्टता प्राप्त करना एवं नवाचार द्वारा उच्च - नवीन मानक स्थापित करना केन्द्रीय विद्यालय संगठन की नियमित कार्यप्रणाली का अविभाज्य अंग है। राष्ट्रीय शिक्षा नीति 2020 एवं पी .एम .श्री विद्यालयों के निर्देशों का पालन करते हुए गतिविधि आधारित पठन-पाठन, अनुभवजन्य शिक्षण एवं कौशल विकास को समाहित कर, अपने विद्यालयों को हमने ज्ञान एवं खोज की अद्‌भुत प्रयोगशाला बना दिया है। माध्यमिक स्तर तक पहुँच कर हमारे विद्यार्थी सैद्धांतिक समझ के साथ-साथ, रचनात्मक - विशेषणात्मक एवं आलोचनात्मक चिंतन भी विकसित कर लेते हैं। यही कारण है कि वह बोर्ड कक्षाओं के दौरान विभिन्न प्रकार के मूल्यांकनों के लिए सहजता से तैयार रहते हैं। उनकी इस यात्रा में हमारा सतत योगदान एवं सहयोग आवश्यक है - केन्द्रीय विद्यालय संगठन के पांचों आंचलिक शिक्षा एवं प्रशिक्षण संस्थान द्वारा संकलित यह विद्यार्थी सहायक -सामग्री इसी दिशा में एक आवश्यक कदम है । यह सहायक सामाग्री कक्षा 9 से 12 के विद्यार्थियों के लिए सभी महत्वपूर्ण विषयों पर तैयार की गयी है। केन्द्रीय विद्यालय संगठन की विद्यार्थी सहायक -सामग्री अपनी गुणवत्ता एवं परीक्षा संबंधी - सामग्री संकलन की विशेषज्ञता के लिए जानी जाती है और शिक्षा से जुड़े विभिन्न मंचों पर इसकी सराहना होती रही है। मुझे विश्वास है कि यह सहायक सामग्री विद्यार्थियों की सहयोगी बनकर निरंतर मार्गदर्शन करते हुए उन्हें सफलता के लक्ष्य तक पहुँचाएगी।

शुभाकांक्षा सहित।

निधि पांडे
आयुक्त, केन्द्रीय विद्यालय संगठन

STUDENT SUPPORT MATERIAL, COMPUTER SCIENCE CLASS XI

PATRON

Smt. Nidhi Pandey, Commissioner, KVS

CO-PATRON

Dr. P. Devakumar, Additional Commissioner (Acad.), KVS (HQ)

CO-ORDINATOR

Ms. Chandana Mandal, Joint Commissioner (Training), KVS (HQ)

COVER DESION

KVS Publication Section

EDITORS:

1. **Ms. Shahida Parveen, Director, ZIET Mumbai**
2. **Mr. B L Morodia, Director, ZIET Gwalior**
3. **Ms. Menaxi Jain, Director, ZIET Mysuru**
4. **Ms. Preeti Saxena, In-charge Director, ZIET Chandigarh**
5. **Mr. Birbal Dhinwa, In-charge Director, ZIET Bhubaneswar**

CONTENT CREATORS:

MRS. SWATI VISHWAKARMA, PRINCIPAL, PM SHRI K V O F CHANDA			
TEAM MEMBERS			
NAME	**DESIGNATION**	**KV NAME**	**REGION**
MRS. SUJA P NAIR	PGT CS	PM SHRI KV ERNAKULAM	ERNAKULUM
MS. VAISHALI D.TITRE	PGT CS	PM SHRI KV NDA KHADAKVASLA	MUMBAI
MRS. RICHA CHAUDHARY	PGT CS	PM SHRI KV SC PUNE	MUMBAI
MRS. SUSHMA CHAUHAN	PGT CS	PM SHRI KV 9 BRD PUNE	MUMBAI
MS. PREETI SARKAR	PGT CS	PM SHRI KV BHANDARA	MUMBAI
MR. DEEP PRAKASH	PGT CS	PM SHRI KV MANDOVI	MUMBAI
MS. RASHMI VERMA	PGT CS	PM SHRI K V O F DEHUROAD	MUMBAI
MRS. VIDYA R PRABHU	PGT CS	PM SHRI KV NO.1 PALAKKAD	ERNAKULUM
MR. SUNIL KUMAR T	PGT CS	PM SHRI KV KELTRON NAGAR	ERNAKULUM

REVIEW TEAM			
NAME	**DESIGNATION**	**KV NAME**	**REGION**
MR. VIVEK SHRIVASTAVA	VICE PRINCIPAL	PM SHRI K V NO. 2 COLABA	MUMBAI
MS. KIRAN SONONE	PGT CS	PM SHRI KV NAD KARANJA	MUMBAI
MR. NIKHIL VERMA	PGT CS	PM SHRI KV NO1 DEHU ROAD	MUMBAI

INDEX OF CONTENTS

COMPUTER SCIENCE
Subject Code - 083
Class XI (2025-26)

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
1	Computer Systems and Organisation	10
2	Computational Thinking and Programming -1	45
3	Society, Law, and Ethics	15
Total		**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, 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 - I

- 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**
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:**
  - $1 + x + x^2 + x^3 + x^4 + \cdots x^n$
  - $1 - x + x^2 - x^3 + x^4 - \cdots x^n$
  - $x + \frac{x^2}{2} + \frac{x^3}{3} + \frac{x^4}{4} + \cdots \frac{x^n}{n}$
  - $x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots \frac{x^n}{n!}$
- 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 integers.
- 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

- A computer is an electronic device that can be programmed to accept data (input), process it and generate result (output). A computer along with additional hardware and software together is called a computer system.
- A computer system primarily comprises a central processing unit (CPU), memory, input/output devices and storage devices. All these components function together as a single unit to deliver the desired output.
- A computer system comes in various forms and sizes. It can vary from a high-end server to personal desktop, laptop, tablet computer, or a smartphone.
- **A computer works on the IPO cycle i.e. Input -> Process -> Output**
- Any kind of computer consists of HARDWARE AND SOFTWARE.

Hardware: Hardware refers to the tangible (physical) components of a computer system.
Input Devices (e.g., Keyboard, Mouse) Output Devices (e.g., Monitor, Printer)
Storage Devices (e.g., Hard Disk, SSD) Processing Unit (e.g., CPU, GPU)

Software: Software is a collection of programs or instructions that operate hardware and perform specific tasks.

Input Devices: These devices are used to **enter data and instructions** into the computer.

Output Devices: The device that receives data from a computer system for display, physical production, etc., is called output device. It converts digital information into human understandable form.

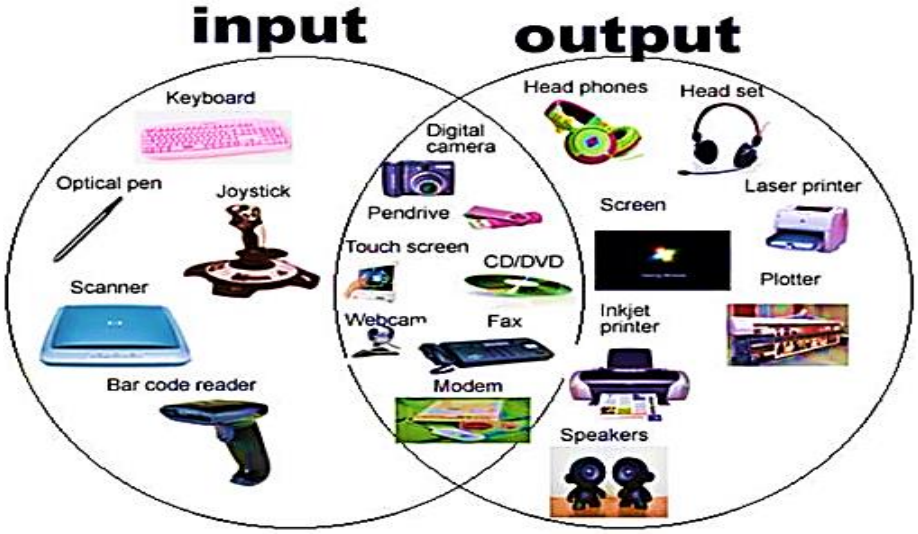


Fig 1

COMPONENTS OF A COMPUTER SYSTEM

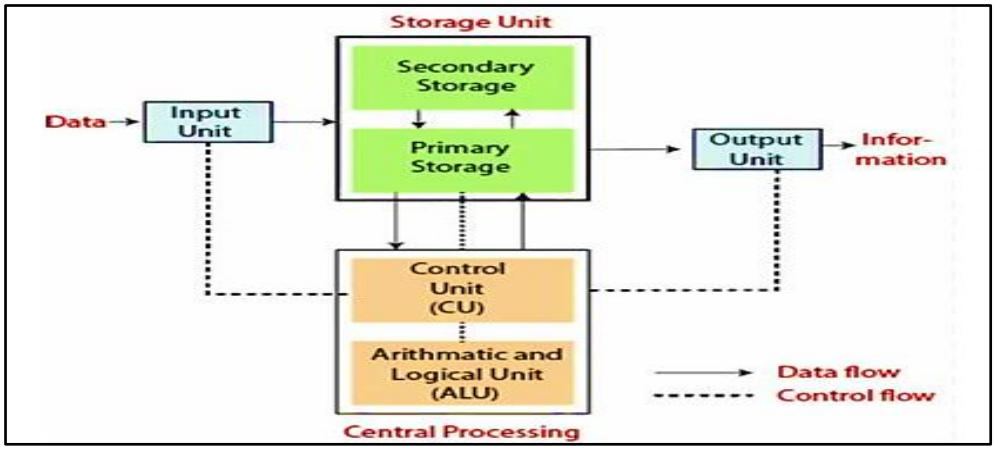


Fig 2

CPU (Central Processing Unit) - It is the electronic circuitry of a computer that carries out the actual processing and is usually referred as the brain of the computer. It is commonly called processor also**.**

Components of CPU:

1. Arithmetic Logic Unit (ALU)
    - Performs arithmetic operations (addition, subtraction, etc.)
    - Performs logical operations (AND, OR, NOT, etc.)
2. Control Unit (CU)
    - Controls all operations inside the CPU
    - Fetches instructions from memory and directs the execution
3. Registers
    - Temporary, fast storage units inside the CPU
    - Store data, instructions, and results during processing

Working of CPU:

- Uses the Fetch-Decode-Execute cycle:
    1. Fetch instruction from memory.
    2. Decode it.
    3. Execute the instruction.

Memory:

Computer memory refers to the electronic components that store data and instructions temporarily or permanently for processing by a computer.

Memory unit is used to store data and instructions.

It stores data in machine language i.e. in the form of 0 and 1. The binary digits 0 and 1 are known as bits.(**BI**nary Digi**TS**)

A) Primary Memory (Main Memory)

Primary memory refers to the memory that is directly accessible by the **CPU**. It stores data and instructions that are currently being used or processed. This memory plays a crucial role in the computer's speed and performance.

➤ **Types of Primary Memory:**

1. **RAM (Random Access Memory):**
    - RAM is a **volatile** memory, which means that it **loses all stored data when the power is turned off**.
    - It is used to **temporarily store data and instructions** that the CPU needs while performing tasks.
    - RAM allows data to be read and written quickly, which is why it is often referred to as **working memory** or **temporary memory**.
    - The more RAM a computer has, the better it can perform multiple tasks at once (**multitasking**).

    Types of RAM:

    DYNAMIC RAM: It consists of capacitors and transistors. It uses electric charge to store the data. It requires constant refreshing to maintain data.

    STATIC RAM: It consists of *flip-flops* (a fundamental digital circuit that stores a single bit of binary data (either 0 or 1)). It has faster access time compared to dynamic RAM.
2. **ROM (Read-Only Memory):**
    - ROM is a **non-volatile** memory, which means it **retains its data even when the power is switched off**.
    - It stores **permanent instructions** that are required to start (boot) the computer, such as the **BIOS (Basic Input Output System)**.
    - As the name suggests, data in ROM can usually only be **read**, not modified (though special types like EEPROM can be updated).

Types of ROM:

PROM (Programmable Read Only Memory): It can be programmed once.

EPROM (Erasable Programmable Read Only Memory): It can be erased by keeping the ROM chip using Ultraviolet light. It can be reprogrammed.
EEPROM (Electrically Erasable Read Only Memory): It can be erased by electrical signal. It can also be reprogrammed

B) Cache Memory: Cache memory is a high-speed memory located inside or very close to the CPU. It stores the most frequently used data and instructions, allowing the CPU to access them more quickly than if it had to fetch them from RAM.

- Cache is much faster than RAM but also **more expensive** and has a **smaller storage capacity**.
- It helps in **reducing the time** taken to access data, thereby improving the overall performance of the computer.

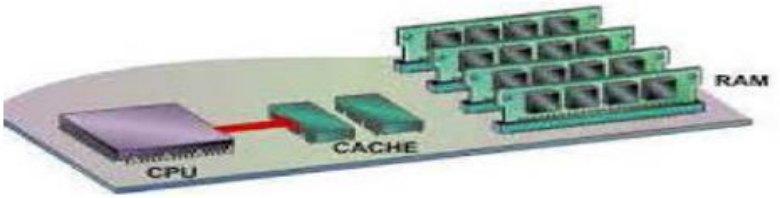


Fig 3

C) Secondary Memory (Storage Devices): Secondary memory is used for the permanent storage of data, files, software, and the operating system. Unlike primary memory, it is non-volatile, meaning data is not lost when the computer is turned off.

➤ Characteristics of Secondary Memory:

- It is not directly accessible by the CPU; data must be loaded into primary memory (RAM) before processing.
- It has large storage capacity compared to primary memory.
- It is slower than primary memory, but suitable for long-term data retention.

➤ Examples of Secondary Storage Devices:

1. Hard Disk Drive (HDD):
   - Magnetic storage device. Stores data in digital form using rotating disks.
2. Solid State Drive (SSD):
   - Uses flash memory. It is faster, more durable, and more expensive than HDDs.
3. Optical Discs (CD/DVD):
   - Use laser technology to read/write data.
4. Flash Drives (Pen Drives, USB drives):
   - Portable, plug-and-play storage devices, Use flash memory
5. Memory Cards (SD Cards):
   - Small in size but capable of storing gigabytes of data.

Fig 4

Data transfer between Memory and CPU:

Data need to be transferred between the CPU and primary memory as well as between the primary and secondary memory.

Data are transferred between different components of a computer system using physical wires called bus.

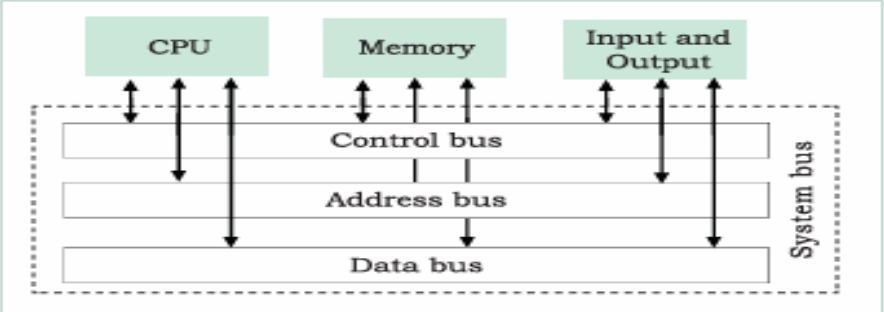


Fig 5

Units of Memory: Memory in a computer system is used to store data and instructions. The amount of data that can be stored in memory is measured using specific units. These units help us understand how much information a memory device can hold.

Basic Unit: Bit

- The **smallest unit** of memory in a computer is a **bit**.
- A bit is short for **Binary Digit**.
- It can hold only **one of two values**:
  - 0 (off, false)
  - 1 (on, true)
- Computers use **binary number system** (base-2), so all data is stored using combinations of bits.

Byte

- **1 Byte = 8 Bits**
- A byte is the **basic unit** of storage used to represent a **single character** (like a letter, number, or symbol).
  - Example: The character **'A'** is stored in **1 byte**.

KB-1024 Bytes	**PB**- 1024 GB
MB- 1024 KB	**EB**- 1024 PB
GB- 1024 MB	**ZB**- 1024 EB
TB- 1024 GB	**YB**- 1024 ZB

TYPES OF SOFTWARE:

Software is defined as a collection of programs which are used for different purposes. There are three types of software:

1) System Software
2) Programming tools and language translators
3) Application Software

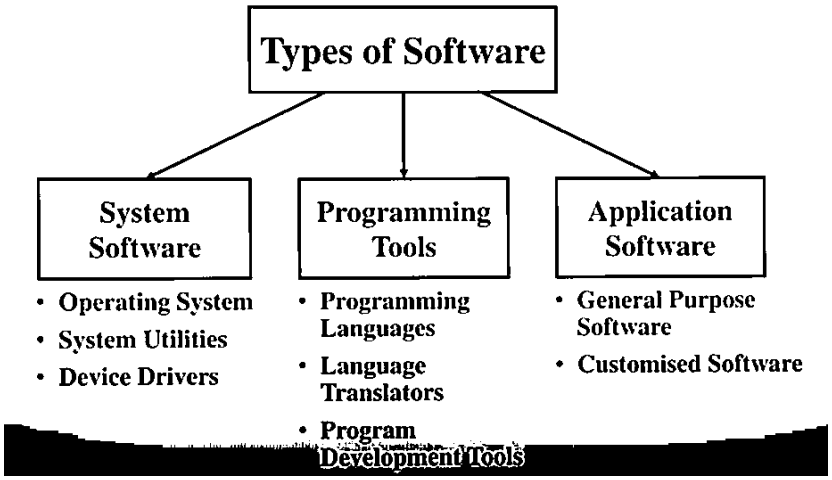


Fig 6

1. System Software

System software is a type of software that controls and manages the hardware components and provides a platform for running application software.

a) Operating System (OS):

- Acts as an interface between user and hardware.
- Manages files, memory, processes, and devices.
- Examples: Windows, Linux, macOS, Android

b) System Utilities:

- Programs that perform maintenance and support tasks.
- Examples: Antivirus, Disk Cleanup, File Compression Tools (WinZip)

c) Device Drivers:

- Specialized software that allows the OS to communicate with hardware devices.
- Each device (printer, keyboard, etc.) needs its own driver.
- Example: Printer Driver, Graphics Driver

2. Programming Tools and Language Translators

a) Programming Language: A programming language is a special language used to write code that tells a computer what tasks to perform.

b) Language Translators: Language translators in computers are programs that convert code written in a programming language into a form the computer can understand and execute. Types of Language Translators:

a) Assembler	b) Compiler	c) Interpreter
Translates assembly language into machine code.	Converts the entire source code (high-level language) into machine code at once.	Translates one line at a time from source code to machine code.
One-to-one conversion.	Fast execution after compilation.	Slower than a compiler.
Used in low-level programming.	Example: C, C++ compilers	Example: Python, JavaScript interpreters

c) Program Development Tools**:** These are tools used by programmers to write, test, debug, and maintain computer programs. Example : IDE(Integrated Development Environment)

3. Application Software

Application software is designed to help users perform specific tasks or activities.

Types:

- General-purpose software – used by most users.
  - Examples: MS Word (word processing), Excel (spreadsheet), PowerPoint (presentation)
- Special-purpose software – developed for a specific task.
  - Examples: Billing Software, Library Management System

Note: Application software runs on top of system software.

OPERATING SYSTEM

An Operating System (OS) is system software that acts as an interface between the user and the computer hardware. It manages all software and hardware resources and provides services for the execution of programs.

FUNCTIONS OF OPERATING SYSTEM	OS USER INTERFACE
• Memory Management	• Command-based Interface

• Process Management • File Management • Device Management • User Interface Management • Security and Access Control • Job Scheduling / Task Management • 8. Error Detection and Handling	• Graphical User Interface • Touch Based Interface • Voice Based Interface • Gesture Based Interface

QUESTION BANK

MULTIPLE CHOICE QUESTIONS

1. Which of the following is the correct order of memory units from smallest to largest?
   a) Byte < Bit < KB < MB < GB b) Bit < Byte < KB < MB < GB
   c) Bit < KB < Byte < MB < GB d) Bit < Byte < MB < KB < GB
2. A student saved a project on a pen drive. Which category of memory does this pen drive belong to?
   a) Primary Memory b) Cache Memory
   c) Secondary Memory d) Virtual Memory
3. Which of the following devices temporarily holds instructions and data that the CPU needs while executing a program?
   a) ROM b) Hard Disk
   c) RAM d) SSD
4. Which type of software is responsible for managing hardware and providing services to other software?
   a) Application Software b) System Software
   c) Compiler d) Word Processor
5. Which of the following correctly matches the language translator with its function?
   a) Assembler – Translates high-level code to binary
   b) Compiler – Executes instructions line by line
   c) Interpreter – Translates and executes code line by line
   d) Compiler – Converts binary to assembly
6. Which of the following is NOT a function of an operating system?
   a) File management b) Application development
   c) Memory allocation d) Process scheduling
7. A user interacts with a computer using a mouse and graphical icons. What type of user interface is this?
   a) Command-Line Interface (CLI) b) Text-Based Interface
   c) Graphical User Interface (GUI) d) Program Interface
8. While working on a large data analysis program, a student notices the program runs slowly due to frequent data access from RAM. Which memory component would help speed up processing in this situation?
   a) Hard disk b) ROM
   c) Cache memory d) USB drive
9. Which of the following best explains why ROM is necessary in a computer system?
   a) It contains the startup instructions for the computer
   b) It speeds up program execution
   c) It stores temporary program data
   d) It manages power supply to components
10. Which of the following is not a system utility (software).
    a) Disk Defragmenter b) Antivirus
    c) Backup Software d) Assembler

ANSWERS

1	2	3	4	5	6	7	8	9	10
b	c	c	b	c	b	c	c	a	d

FILL IN THE BLANKS

1. Raw facts and figures are called_____.
2. Processing takes place in the part of the computer known as the _____.
3. The _____ directs and coordinates all the activities within the CPU.
4. A _____ is the sequence of instructions performed to solve one problem.
5. _____ holds data and instructions that the computer is processing at the time.

ANSWERS

1	2	3	4	5
Data	CPU (Central Processing Unit)	Control Unit	Algorithm	Primary memory (RAM)

ASSERTION AND REASONING QUESTIONS

In the following questions, a statement of Assertion(A) is followed by a statement of Reason (R). Make 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 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

1. **Assertion (A):** The system software is set of programs which takes care of all the activities of a computer system.
   Reason (R): The system software is designed by the manufacturers in low level languages to serve as an interface between the user and the computer. It controls the various hardware and software components of a computer system.
2. **Assertion(A):** An operating system is a system software that manages various resources and the overall operations of a computer system.
   Reason(R): An operating system can be classified into two categories: single-user and multi-user operating system. A single-user operating system allows only one user whereas, the multi-user operating system allows more than one user to interact with the computer at a time.
3. **Assertion(A):** An operating system is a system software that manages various resources and the overall operations of a computer system.
   Reason(R): An operating system coordinates different hardware and software components of a computer system. It also helps in the smooth functioning of various peripherals.
4. **Assertion(A):** Compiler and interpreter are the two important language translators used in computer system.
   Reason(R): The compiler is the software that accepts the entire program in high level language and converts into its equivalent machine language at once whereas, the interpreter converts the instructions into its equivalent machine language line by line.
5. **Assertion(A):** An application software is designed to run the computer system smoothly and effectively.

Reason(R): An application software is the set of programs to fulfil the specific needs of the user. The Microsoft Office package includes some well-known application software such as MS- Word, MS-Excel, MS-Power Point, MS-Access to perform specific task.

ANSWERS

1	2	3	4	5
A	B	A	A	D

SHORT ANSWER TYPE QUESTIONS

Q1. What is the role of the CPU in a computer system?
Q2. What is cache memory? How is it different from RAM?
Q3. What is an interpreter? How is it different from a compiler?
Q4. List two differences between primary memory and secondary memory.
Q5. What is the function of device drivers in a computer system?
Q6. Explain the term 'bit' and 'byte'.
Q7. What is the role of the operating system in process management?
Q8. Name any two system utilities and their uses.
Q9. What is a boot loader?
Q10. What do you understand by IPO cycle?

ANSWERS

Ans 1. The CPU (Central Processing Unit) is the brain of the computer. It performs all arithmetic and logical operations, controls instructions, and manages the flow of data within the system. It consists of the ALU (Arithmetic Logic Unit), CU (Control Unit), and Registers.
Ans 2. Cache memory is a small, high-speed memory located near or inside the CPU that stores frequently accessed data. It is faster than RAM but smaller in size. RAM is used for general temporary storage while a program runs.
Ans 3. An interpreter translates high-level code line-by-line, whereas a compiler translates the entire program at once into machine code. Interpreters are slower but allow real-time error checking.
Ans 4. Primary memory is faster and volatile (e.g., RAM), while secondary memory is slower and non-volatile (e.g., HDD).
Primary memory is used by the CPU during execution, secondary memory is used for long-term storage.
Ans 5. Device drivers are system software that enable the operating system to communicate with hardware devices like printers, scanners, or keyboards.
Ans 6. A bit (binary digit) is the smallest unit of data and can be either 0 or 1.
A byte consists of 8 bits and represents one character (like a letter or number).
Ans 7. The OS manages execution of multiple processes, schedules CPU time, and ensures smooth multitasking by switching between programs efficiently.
Ans 8. Antivirus software – scans and removes viruses.
Disk Cleanup – removes temporary and unused files to free up space.
Ans 9. A Boot Loader (also called a Boot Manager or Boot Program) is a special program that loads the operating system (OS) into memory when the computer is turned on or restarted. It is stored in a special part of the system's ROM or in the Master Boot Record (MBR) of the hard disk.
Ans 10. IPO cycle refers to the Input Process Output cycle where every operation undergoes the phases namely input, process and output.

CASE STUDY BASED QUESTION

Taniya is a beginner of using software. She opened a start-up company to do business. She wants to purchase some software for writing reports, saving employees data, mathematical calculations, watching videos, listening music, presentation in meeting, real time communication with audio & video and email. She is confused to take proper decision for few applications software. Being a well-wisher of Taniya, what will you suggest her to purchase the software for the said purposes.

(i) Writing report in office.
(ii) Database software to save employee data.
(iii) Software for mathematical calculations.
(iv) Software for real time communication with audio & video.
(v) Application software for email communication.

ANSWERS

i) Ans: Microsoft Word, Google Docs, WordPad and Notepad
ii) Ans: Microsoft Access, MySQL, Microsoft SQL Server
iii) Ans: Google Sheets, Apple Numbers and Microsoft Excel
iv) Ans: Google Meet, Zoom, Skype
v) Ans: Gmail, Microsoft Outlook, Apple Mail

LONG ANSWER TYPE QUESTIONS

Q1. Explain the different types of memory in a computer system.
Q2. What are the key functions of an operating system? Explain any four.
Q3. Compare and contrast assembler, compiler, and interpreter.
Q4. Describe the components of a computer system with examples and Diagram.
Q5. Differentiate between CLI and GUI. Give examples of OS that use each.

ANSWERS

Ans1. 1. Primary Memory (Main Memory)

Primary memory is the main area where the computer stores data and instructions that are currently being used. It is directly accessible by the CPU and is typically volatile, meaning it loses its contents when power is turned off. Primary memory mainly includes:

- RAM (Random Access Memory): Temporary storage for active processes and data. It allows both read and write operations and is essential for system performance.
- ROM (Read-Only Memory): Contains permanent instructions such as firmware. It is non-volatile and cannot be easily modified.

2. Secondary Memory (Storage Devices)

Secondary memory refers to non-volatile, long-term storage. Unlike primary memory, it retains data even when the computer is turned off. It is used to store the operating system, software applications, and user files.

- Common types include:
  Hard Disk Drives (HDD) Solid State Drives (SSD)
  Optical disc (CD/ DVD) USB flash drives

3. Cache Memory

Cache memory is a small, high-speed memory located close to the CPU. It stores frequently accessed data and instructions to speed up processing. Because it's much faster than RAM, it helps reduce the time the CPU takes to fetch data.

- It is typically divided into levels: L1, L2, and L3, with L1 being the fastest and smallest.

Ans2. The operating system (OS) is system software that manages computer hardware, software resources, and provides services for computer programs. It acts as an interface between the user and the hardware.

Here are four key functions of an operating system, each explained clearly:

1. Process Management
The OS manages processes in a system, including their creation, scheduling, execution, and termination. It handles multitasking by allocating CPU time to various processes and ensures efficient execution by prioritizing tasks.
2. Memory Management
The OS manages the computer's physical and virtual memory. It keeps track of each byte in a computer's memory and allocates or deallocates memory spaces as needed by programs.
3. File System Management
The OS manages files on storage devices. It handles operations like creation, deletion, reading, writing, and organization of files in directories.
4. Device Management
The OS manages input and output devices such as keyboards, mice, printers, and monitors through drivers. It acts as a bridge between hardware and software by using device drivers.
Ans 3.

Tool	Definition	Purpose
Assembler	Converts assembly language into machine code.	Bridges human-readable assembly and binary machine language.
Compiler	Translates high-level language (e.g., C, Java) into machine code.	Converts entire source code into an executable file before running.
Interpreter	Translates and executes high-level code line-by-line.	Allows immediate execution of code without creating an executable file.

Ans 4.
Input Unit: Devices or components that send data and instructions into the computer (e.g., keyboard, mouse).
Primary Memory: Fast, temporary memory that stores data and programs currently in use (e.g., RAM).
Secondary Memory: Permanent storage for data and programs, even when the computer is off (e.g., hard drives).
Control Unit: Part of the CPU that directs all operations by managing instructions and data flow.
ALU (Arithmetic Logic Unit): CPU component that performs arithmetic and logical operations.
Output Unit: Devices or components that display or deliver processed data to the user (e.g., monitor, printer).
Draw Components Of A Computer System(Fig2)

Ans5.

Feature	CLI (Command Line Interface)	GUI (Graphical User Interface)
What it looks like	You type text commands	You click pictures, icons, and buttons
How you use it	Write commands on the keyboard	Use a mouse or touch to click and select
Easy or hard?	Harder for beginners, you need to learn commands	Easy and simple to use, good for everyone
Speed	Fast if you know commands	Slower because of graphics
Uses more resources?	Uses less computer power	Uses more computer power
Examples of Operating Systems	Linux (using Terminal), MS-DOS	Windows, macOS, Ubuntu (Desktop version)

BOOLEAN LOGIC

Boolean logic is a branch of algebra that deals with true/false **values**, represented as **1 (True) and 0 (False)**. It is the basis of digital circuits and computer operations. It was developed by the English Mathematician and logician George Boole. Boolean Algebra comprises of following:

1. Boolean Expression
2. Boolean Variable

Boolean Expression

A boolean expression is an expression that consists of a combination of boolean variables, boolean values and boolean operators *(such as NOT, AND, OR)*. A boolean expression evaluates to either True or False.

- Logic Statement - a sentence that is either true or false, but not both.

Boolean Variable

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

Truth Table

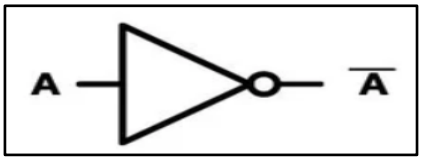
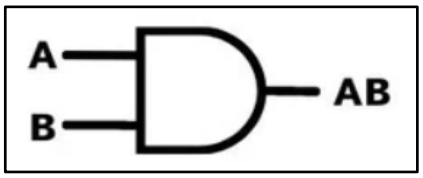
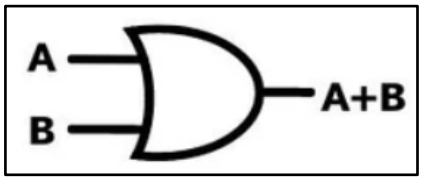
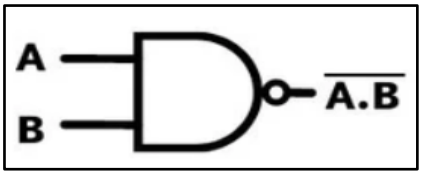
A truth table is a representation of all possible combinations of the input variables and the corresponding output values

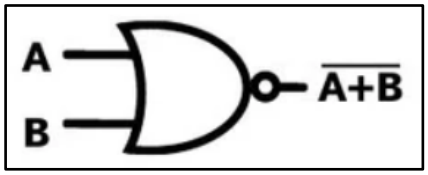
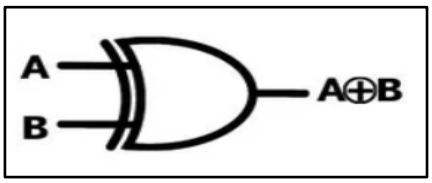
The number of rows in a truth table are 2^n, where n is no. of input variables.

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

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

Logic Gates - Logic gates are electronic circuits that perform Boolean logic operations on one or more binary inputs (0 or 1) to produce a single binary output. They are basic building blocks of digital circuits.

GATE	SYMBOLS AND NOTATIONS	TRUTH TABLE
NOT GATE **Also called Inverter*		A | A' 0 | 1 1 | 0
AND GATE		A | B | A.B 0 | 0 | 0 0 | 1 | 0 1 | 0 | 0 1 | 1 | 1
OR GATE		A | B | A+B 0 | 0 | 0 0 | 1 | 1 1 | 0 | 1 1 | 1 | 1
NAND GATE **NOT AND*		A | B | (A.B)' 0 | 0 | 1 0 | 1 | 1 1 | 0 | 1 1 | 1 | 0

NOR GATE **NOT OR*		A | B | (A+B)' 0 | 0 | 1 0 | 1 | 0 1 | 0 | 0 1 | 1 | 0
XOR GATE **eXclusive OR* *Gives high when odd numbers of input are high*		A | B | A⊕B 0 | 0 | 0 0 | 1 | 1 1 | 0 | 1 1 | 1 | 0

De Morgan's Laws

De Morgan'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.

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

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

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

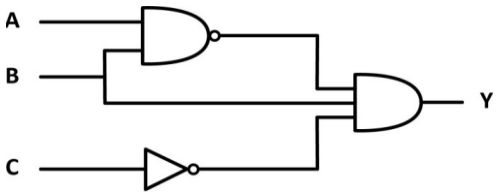
(A+B)'=A'. B'

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

Logic Circuits

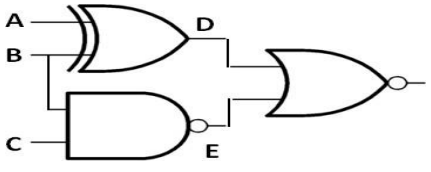
A logic circuit is an electronic circuit which performs logical operations on the input Boolean variables, and transforms them into the output using a combination of Logic Gates.

Examples of a circuit diagram are:



Y = (A.B)'.BC'

Do it Yourself



Y = ?

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	A.B=B.A	A+B=B+A
Associative Law	(A.B).C=A.(B.C)	(A+B)+C=A+(B+C)
Distributive Law	A+B.C=(A+B)(A+C)	A.(B+C)=A.B+A.C
Absorption Law	A.(A+B)=A	A+A.B=A

QUESTION BANK

MULTIPLE CHOICE QUESTIONS

1. The output of the two input OR gate is high when
   a) Both inputs are low b) Both inputs are high
   c) Any one input is high d) Only one input is high
2. The output of the two input AND gate is high when
   a) Both inputs are low b) Both inputs are high
   c) Any one input is high d) Only one input is high
3. Which of the following is/are the universal logic gates?
   a) OR and NOR b) AND c) NAND and NOR d) NOT
4. A Boolean Expression which always results in True is called_____.
   a) True b) Tautology c) Fallacy d) False
5. The output of the two input XOR gate is high when
   a) Both inputs are low b) Both inputs are high
   d) Any one input is high d) Only one input is high
6. To denote NOT operation which of the following symbols is used?
   a) asterisk b) dot
   c) plus d) bar
7. The dot symbol is used to represent which of the following operation?
   a) AND b) OR
   c) NOR d) XOR
8. Which of the following are logical statements:
   a) The Indian cricket team is a world champion in 2011.
   b) The NOT operator can take more than one boolean value as input.
   c) Both a & b.
   d) None of the above
9. Which of the following gate is also known as an inverter
   a) AND b) OR
   c) NOT d) NAND
10. Which two input gate(s) returns true if no input is high.
    a) NAND b) NOR
    c) XOR d) None of the above

ANSWERS

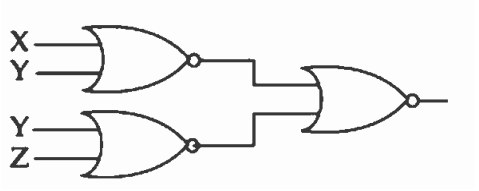
1	2	3	4	5	6	7	8	9	10
c	b	c	b	d	d	a	c	c	a & b

SHORT ANSWER TYPE QUESTIONS

Q1. What is a truth table? What is its significance?

Q2. What is meant by tautology and fallacy? Prove that 1+Y is a tautology and 0.Y is a fallacy.

Q3. Prove the complementarity law of boolean algebra with the help of a truth table.
Q4. Draw logical circuit for the following equation:
a) A.(B+C') b)A.B' + C'
Q5. Write the equivalent Boolean expression for the following logic circuit.



ANSWERS

Ans 1. A Truth Table is a table which represents all the possible values of logical variables / statements along with all the possible results of the given combinations of values.

Ans2. **Tautology**: - If the result of any logical statement or Boolean expression is always TRUE or 1 for all input combinations, it is called Tautology.

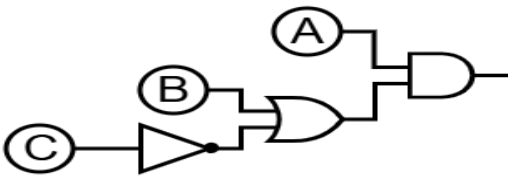
Fallacy: - If the result of any logical statement or expression is always FALSE or 0 for all input combinations, it is called Fallacy.

- If we put value of Y = 0 or 1 in 1+Y, then result is always 1 that's why this is tautology.
- If we put value of Y = 0 or 1 in 1.Y, then result is always 1 that's why this is fallacy

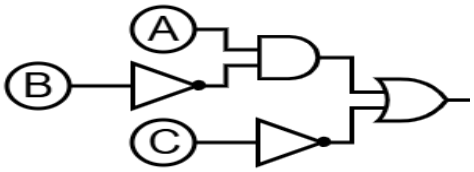
Ans 3. (i) Truth Table for X + X' = 1 (ii)Truth Table for X.X' = 0

X+X'			X.X'		
X	**X'**	**X + X'**	**X**	**X'**	**X.X'**
0	1	1	0	1	0
1	0	1	1	0	0

Ans 4. a)



b)

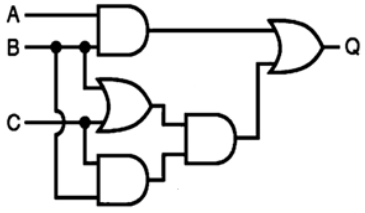


Ans 5.

$((X+Y)' + (Y+Z)')'$ or $\overline{\overline{X+Y} + \overline{Y+Z}}$

LONG ANSWER TYPE QUESTIONS

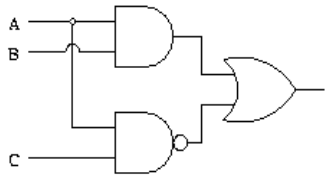
Q. 1 Draw a logic circuit for the following Boolean expression: **A.B + (A.C)'**
Q. 2 Write the equivalent boolean expression for the following logic circuit.



Q3. State and Prove De Morgan's law using Truth Table.

ANSWERS

Ans 1.



Ans 2. **Q=AB+(B+C).(BC)**

Ans 3.

Demorgan's First Law – **$(A.B)'=A'+B'$**

A	B	$\overline{A+B}$	$\overline{A}$	$\overline{B}$	$\overline{A}.\overline{B}$
0	0	1	1	1	1
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	0

Demorgan's Second Law - **$(A+B)'=A'.B'$**

A	B	$\overline{A+B}$	$\overline{A}$	$\overline{B}$	$\overline{A}+\overline{B}$
0	0	1	1	1	1
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	0	0	0

NUMBER SYSTEM

A **number system** is a way to represent and express numbers using a specific set of **symbols (digits)** and a **base (or radix)**. It defines how numbers are written, read, and interpreted in mathematics and computing.

Each number system has:

- A **base (radix)**: the total number of unique digits, including zero.
- A **set of digits**: the symbols used to represent numbers.
- A **positional value system**: the position of a digit determines its value (e.g., in decimal, 543 = 5×100 + 4×10 + 3×1).

Types of Number Systems

1. Binary Number System (Base-2)

- **Digits used**: 0 and 1
- Used internally by almost all **modern computers and digital systems**
- Each digit is called a **bit**
- **Example**:
  $(1011)_2 = (1\times2^3) + (0\times2^2) + (1\times2^1) + (1\times2^0) = 8+0+2+1 = 11_{10}$

2. Octal Number System (Base-8)

- **Digits used**: 0 to 7
- Common in early computing
- **Shortcut**: 3 binary digits = 1 octal digit
- **Example**:
  $(17.5)_8 = (1\times8^1) + (7\times8^0) + (5\times8^{-1}) = 8+7+(5\times0.125) = 15+0.625 = (15.625)_{10}$

3. Decimal Number System (Base-10)

- **Digits used**: 0 to 9
- Most commonly used system in daily life
- **Example**:
  $347_{10} = (3\times100) + (4\times10) + (7\times1)$

4. Hexadecimal Number System (Base-16)

- **Digits used**: 0 to 9 and A to F (A=10, B=11, ..., F=15)
- Widely used in **computer memory addressing** and **color codes in HTML/CSS**
- **Shortcut**: 4 binary digits = 1 hex digit
- **Example**:
  $2F.A_{16} = (2\times16^1) + (15\times16^0) + (10\times16^{-1}) = 32+15+(10\times0.0625) = 47+0.625 = (47.625)_{10}$

Conversion Between Number Systems

1. Binary ↔ Decimal

- **Binary to Decimal**: Multiply each bit by 2^n (from right to left) and add.
- Digits **before the binary point** use powers of $\mathbf{2^n}$ ($n \geq 0$).
- Digits **after the binary point** use **negative powers of 2** (2^{-1}, 2^{-2}, 2^{-3}, ...).

Example: Convert $\mathbf{101.101_2}$ to decimal

Split it into:

- **Integer part**: $101_2 = 1\times2^2+0\times2^1+1\times2^0 = 4+0+1 = 5$
- **Fractional part**: $.101_2 = 1\times2^{-1}+0\times2^{-2}+1\times2^{-3} = 0.5+0+0.125 = 0.625$

$$\mathbf{101.101_2 = 5 + 0.625 = 5.625_{10}}$$

2. Binary ↔ Octal

- **Binary to Octal**: Group binary digits in **3s from right**, then convert each group.
  - Example: 110101 → 110 101 = 6 5 → 65_8
  - Example : 10101.1101 = **0**10 101 . 110 **100** = 2 5 . 6 4 = 25.64_8
- **Octal to Binary**: Convert each octal digit to **3-bit binary**.
  - Example: 73_8 → 111 011 → 111011_2

- Example: $12.36_8 \rightarrow \underline{001}\ \underline{010}\ .\ \underline{011}\ \underline{110} \rightarrow 1010.01111_2$

3. Binary ↔ Hexadecimal

- **Binary to Hex**: Group binary digits in **4s from right**, then convert each group.
  - Example: $11011110_2 \rightarrow \underline{1101}\ \underline{1110} = D\ E \rightarrow DE_{16}$
  - Example: $101.110_2 \rightarrow \mathbf{0}\underline{101}\ .\ \underline{1100} \rightarrow 5\ C \rightarrow 5C_{16}$
- **Hex to Binary**: Convert each hex digit to **4-bit binary**.
  - Example: $A3_{16} \rightarrow \underline{1010}\ \ \underline{0011} \rightarrow 10100011_2$
  - Example: $2B.F_{16} \rightarrow \underline{0010}\ \underline{1011}\ .\ \underline{1111} \rightarrow 101011.1111_2$

4. Decimal ↔ Other Bases

- **Decimal to Binary/Octal/Hex**: Use repeated division by 2, 8 or 16 and write down remainders (bottom to top).
- **Converting 125_{10}** into Binary, Octal and Hexadecimal

Binary

Division	Quot	Rem
125 ÷ 2	62	1
62 ÷ 2	31	0
31 ÷ 2	15	1
15 ÷ 2	7	1
7 ÷ 2	3	1
3 ÷ 2	1	1
1 ÷ 2	0	1

$125_{10} = 1111101_2$

Octal

Division	Quot	Rem
125 ÷ 8	15	5
15 ÷ 8	1	7
1 ÷ 8	0	1

$125_{10} = 175_8$

Hexadecimal

Division	Quot	Rem
125 ÷ 16	7	13
7 ÷ 16	0	7

13 = D in hexadecimal.

$125_{10} = 7D_{16}$

Encoding Schemes

1. ASCII (American Standard Code for Information Interchange)

- 7-bit encoding (128 characters)
- Represents **English characters, digits, punctuation, and control codes**
- Example:
  - 'A' = 65, 'a' = 97, '0' = 48 (in decimal)

2. ISCII (Indian Script Code for Information Interchange)

- 8-bit encoding
- Designed for **Indian scripts** like Devanagari, Tamil, Bengali, etc.
- Superset of ASCII (first 128 values same)
- Used for multilingual computing in India

3. Unicode

- A **universal character encoding standard** that supports almost every written script.
- Supersedes ASCII and ISCII
- Common encodings:
  - **UTF-8**: Variable-length (up to 4 blocks of 8 bits), backward compatible with ASCII. Most commonly used.
  - **UTF-32**: Fixed-length (32 bits), uses more memory but faster lookup.
- Example:
  - 'A' = U+0041, 'अ' (Devanagari) = U+0905

QUESTION BANK

MULTIPLE CHOICE QUESTIONS

1. An organized way of representing numbers is called:

a) Character System b) Number system c) Integer system d) None

2. How graphics are represented by the computer?
   a) Unicode b) ASCII c) Bitmap d) Binary
3. Using ASCII how many bits are required for character 'A'?
   a) 11 b) 7 c) 8 d) 14
4. Which of the following is not an example of Octal number?
   a) 456 b) 370 c) 78 d) 100
5. Which of the following symbol is not a part of Hexadecimal number system?
   a) 0 b) 9 c) 10 d) A
6. 4 bit binary code for decimal number 15 is……
   a) 1100 b) 1101 c) 1110 d) 1111
7. Binary equivalent of decimal number 32 is… ?
   a) 10000 b) 100000 c) 1000 d) 100
8. Which of the following will occupy more memory pace?
   a) A2 b) 12 c) 10 d) 1111
9. Octal number equivalent to binary number 1110101 is… ?
   a) 115 b) 165 c) 65 d) 185
10. $(1101011.10101)_2$ can be represented in Hexadecimal number as?
   a) 6B.68 b) 6B. A8 c) 6B. A81 d) 6A.A8
11. What is the decimal equivalent of the binary number 101001?
   a) 41 b) 37 c) 45 d) 33
12. What is the hexadecimal equivalent of the octal number 7652?
   a) 1F5 b) 1E5 c) FAA d) FC5
13. What is the octal number 20 expressed in the decimal system?
   a)16 b) 18 c) 20 d) 22
14. What is the binary equivalent of the hexadecimal number 2F2?
   a) 001011110010 b) 010011111111
   c) 001111110101 d) 010111111110
15. What is the octal number 4307 expressed in the decimal system?
   a) 2247 b) 2347 c) 2447 d) 2547

ANSWERS

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
b	c	b	c	c	d	b	a	b	b	a	c	a	a	a

TRUE OR FALSE QUESTIONS

1. Binary is a base-2 number system.
2. Decimal is a base-8 number system.
3. Octal uses 16 distinct digits.
4. Hexadecimal uses symbols 0 to 15.
5. In binary, each digit represents a power of 2.

ANSWERS

1	2	3	4	5
True	False	False	False	True

ASSERTION AND REASONING QUESTIONS

In the following questions, a statement of Assertion(A) is followed by a statement of Reason (R). Make 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 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

Q1) **Assertion:** The binary number 10101010 can be written as AA in hexadecimal.
Reason: The hexadecimal number system uses 16 symbols, from 0 to 9 and A

to F, to represent Numbers.

Q2) **Assertion**: The octal number system has a base value of 8.
Reason: The octal number system uses 8 digits, from 0 to 7, to form numbers.

Q3) **Assertion**: ASCII can represent characters from all writing systems around the world.
Reason: ASCII is a widely used character encoding scheme in computing.

Q4) **Assertion:** ISCII is primarily used to encode characters from non-Indian languages.
Reason: ISCII includes additional characters for Indian scripts.

Q5) **Assertion**: UTF-8 can represent all characters in the Unicode standard.
Reason: UTF-8 uses variable-length encoding.

ANSWERS

1	2	3	4	5
A	A	D	D	A

SHORT ANSWER TYPE QUESTIONS

1. Expand the following terms: ASCII, ISCII.
2. Explain octal and hexadecimal number system.
3. Explain decimal and binary number system.
4. Convert the following binary numbers to decimal:
   (a) 111010 (b) 101011111

ANSWERS

Ans 1. ASCII: American Standard Code for Information Interchange.
ISCII: Indian Script code for Information Interchange.

Ans 2.

- Octal (base 8) was previously a popular choice for representing digital circuit numbers in a form that is more compact than binary. Octal is sometimes abbreviated as oct. Octal counting goes as: 0,1, 2, 3,4, 5, 6, 7,10,11,12,13,14,15,16,17,20, 21 and so on.
- Hexadecimal (base 16) is currently the most popular choice for representing digital circuit numbers in a form that is more compact than binary. Hexadecimal numbers are sometimes represented by preceding the value with '0x', as in 0x1B84. Hexadecimal is sometimes abbreviated as hex. Hexadecimal counting goes: 0,1,2, 3,4,5, 6, 7, 8,9, A, B, C, D, E, F,10 and so on.

Ans 3. Decimal (base 10) is the way most human beings represent numbers. Decimal is sometimes abbreviated as dec. Decimal counting goes: 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18 and so on.

- Binary (base 2) is the natural way most digital circuits represent and manipulate numbers. Binary numbers are sometimes represented by preceding the value with '0b', as in 0b1O11. Binary is sometimes abbreviated as bin. Binary counting goes as: 0,1,10,11,100,101,110, 111, 1000, 1001, 1010,1011, 1100, 1101, 1110, 1111,10000,10001 and so on.

Ans 4 :-

(a)

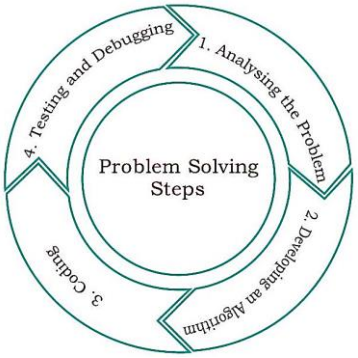
1	1	1	0	1	0	
2^5	2^4	2^3	0	2^2	0	$(111010)2=(60)_{10}$

(b)

1	0	1	0	1	1	1	1	1	
2^8	0	2^6	0	2^4	2^3	2^2	2^1	1	$(101011111)2=(351)_{10}$

COMPUTATIONAL THINKING AND PROGRAMMING – I
INTRODUCTION TO PROBLEM-SOLVING

- Problem solving is the **act of defining a problem**; determining the cause of the problem; identifying, prioritizing, and selecting alternatives for a solution; and implementing a solution.
- In programming, this means taking a real-world issue or task and finding a way to solve it using a computer program. Effective problem-solving is essential for writing efficient, accurate, and maintainable code.



STEPS OF PROBLEM SOLVING

1. **Analyse The Problem-** It is important to clearly understand a problem before we begin to find the solution for it. By analyzing a problem, we would be able to figure out what are the inputs that our program should accept and the outputs that it should produce.
2. **Algorithm** – A set of exact steps which when followed, solve the problem or accomplish the required task.
   EXAMPLE 1- Algorithm to prepare Tea
   1) boil water
   2) add tea (bag or loose leaf)
   3) add any desired ingredients like milk, sugar, and spices
   4) brew the tea
   5) serve and enjoy.

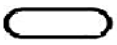
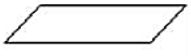
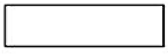
EXAMPLE 2 - Algorithm to calculate area of circle with radius greater than 20

1. Take input for radius from user in float
2. If radius is greater than 20 then go to step 3 otherwise go to step 1
3. Write the formula for area of circle. Area = (22/7) * radius * radius
4. Print the output – Area

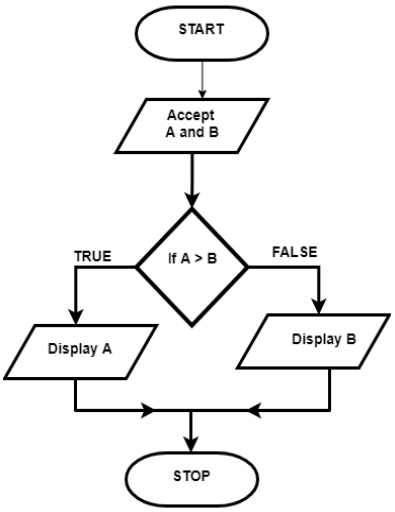
Flowchart — Visual Representation of Algorithms

- A flowchart is a visual representation of an algorithm. A flowchart is a diagram made up of boxes, diamonds and other shapes, connected by arrows. Each shape represents a step of the solution process and the arrow represents the order or link among the steps OR FLOW OF CONTROL OF THE PROGRAM.

FLOWCHART SYMBOLS

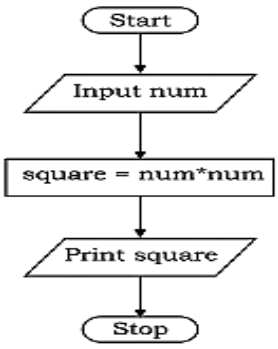
Symbol	Symbol Name	Description
	Flow Lines	Used to connect symbols
	Terminal	Used to start, pause or halt in the program logic
	Input/output	Represents the information entering or leaving the system
	Processing	Represents arithmetic and logical instructions
	Decision	Represents a decision to be made
	Connector	Used to Join different flow lines

Example 1. Flowchart to find largest of two numbers



Example 2. Write an algorithm to find the square of a number and draw flowchart of it.

Algorithm to find square of a number.
Step 1: Input a number and store it to num
Step 2: Compute num * num and store it in square
Step 3: Print square



PSEUDOCODE

A pseudocode (pronounced Soo-doh-kohd) is another way of representing an algorithm. It is considered as a non-formal language that helps programmers to write algorithm. It is a detailed description of instructions that a computer must follow in a particular order.
There is no strict set of standard notations for pseudocode, but some of the most widely recognised are:
INPUT/READ – indicates a user will be inputting something
OUTPUT/WRITE – indicates that an output will appear on the screen
WHILE – a loop (iteration that has a condition at the beginning)
FOR – a counting loop (iteration)
REPEAT – UNTIL – a loop (iteration) that has a condition at the end.
IF – THEN – ELSE – a decision (selection) in which a choice is made

EXAMPLE 1 - Pseudocode to find out largest of two numbers

START
INPUT n1, n2
IF (n1 > n2) THEN
Set max to n1 Else
Set max to n2

EXAMPLE 2 - Pseudocode to print table of 5

START
SET number = 5
FOR i FROM 1 TO 10 DO
  PRINT number × i
END FOR
END

3. CODING

After finalising the algorithm, we need to convert the algorithm into the format which can be understood by the computer to generate the desired solution. Different high level programming languages can be used for writing a program, such as FORTRAN, C, C++, Java, Python, etc.

```
import  math
rad= float (input('Enter radius in float'))
Area= math.pi*rad*rad     # or 3.14*rad*rad
print('Area of Circle :',Area)
```

4. TESTING AND DEBUGGING

TESTING is the process of checking if a program **works correctly** and **meets requirements**.

Importance of Testing:

- Saves time and cost by finding bugs early.
- Increases software reliability and user satisfaction.

DEBUGGING

Debugging is the process of **finding and fixing** errors (bugs) discovered during testing.

DECOMPOSITION

Decomposition means **breaking down a big problem into smaller, easier parts**.

Use of Decomposition in programming

- Easier to solve small problems than one big one.
- Helps organize tasks clearly.
- Makes code cleaner and easier to debug.

EXAMPLE 1- Organize **a birthday party.**

Decomposed into smaller tasks:

1. Choose a date and time.
2. Make a guest list.
3. Decide on a location.
4. Plan the food and drinks.
5. Buy decorations and party supplies.
6. Send invitations.
7. Set up and decorate.
8. Host the party.

EXAMPLE 2- To build a calculator app.

We can **decompose** the problem as:

- Input number and operations from user
- Add two numbers
- Subtract two numbers
- Multiply two numbers
- Divide two numbers
- Displaying Results

BASICS OF PYTHON PROGRAMMING

Python is a high-level, interpreted programming language known for its simplicity and readability. It was created by **Guido van Rossum** and first released in **1991**.
Python is widely used in:

- Web development
- Data science
- Machine learning
- Automation
- Software development

Python Pluses / Features of Python

- Object Oriented Language
- Simple and Easy to Learn
- Expressive Language
- Interpreted Language
- Dynamically Typed
- Extensive Standard Library
- Its Completeness in library – *"Batteries included"*
- Cross Platform Language
- Free and Open Source

How easy and expressive python is, can be understood by this example-

If we want to display a simple *Hello World* message on screen using python, we have to execute the following command

print("HELLO WORLD!!")

isn't this easy and expressive.

Execution Modes in Python

There are two modes to work on python:

- Interactive (also called Immediate Mode)
- Script Mode

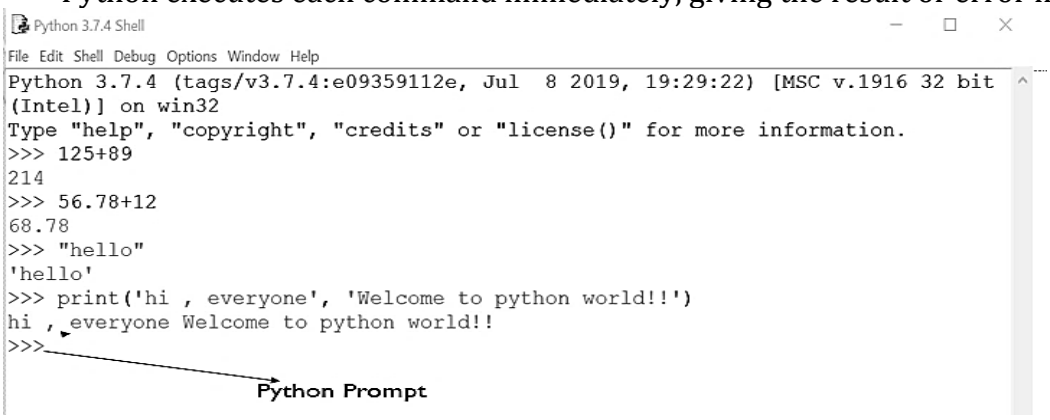
Working in Interactive Mode

Interactive mode means you type the command – one command at a time and the Python executes the given command there and then and gives you the output
In Interactive mode, the commands are given in front of Python command prompt >>> (Angular bracket).

Example

```
>>> 12+ 13
      25
>>> print('Hello')
     Hello
```

Interactive mode is **useful for testing code**. You type commands one by one, and Python executes each command immediately, giving the result or error instantly.

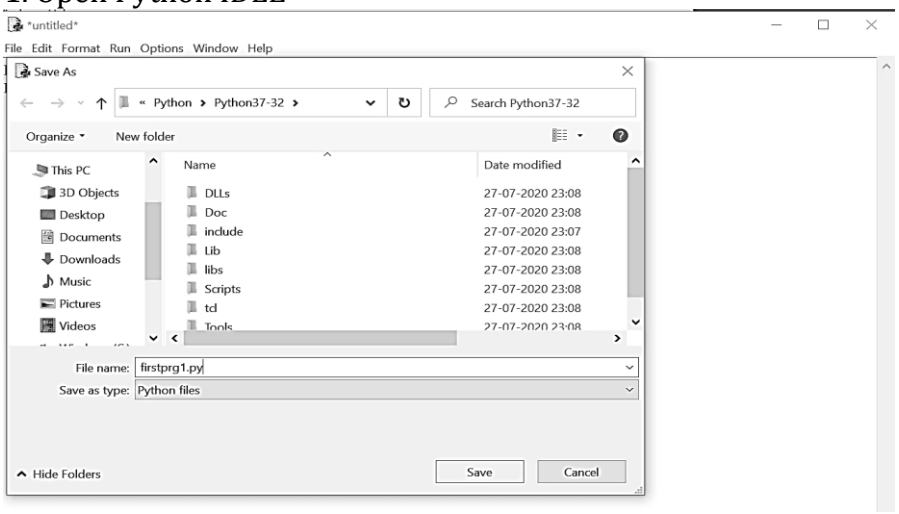


Working in Script Mode -

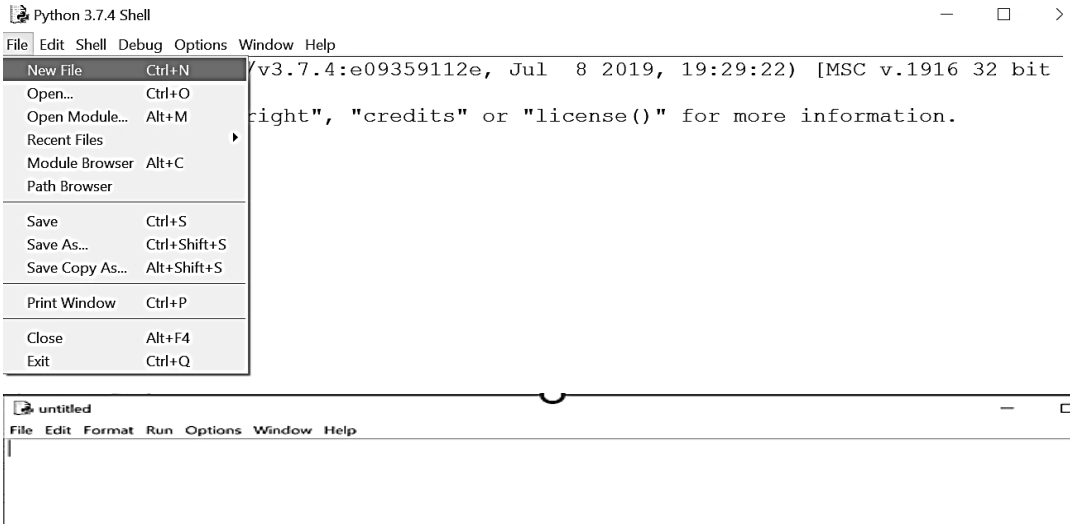
Script Mode is used when we want to save all the commands in the form of program file and want to see all output lines together rather than sandwiched between successive commands.

Steps for writing in Script mode –

1. Open Python IDLE

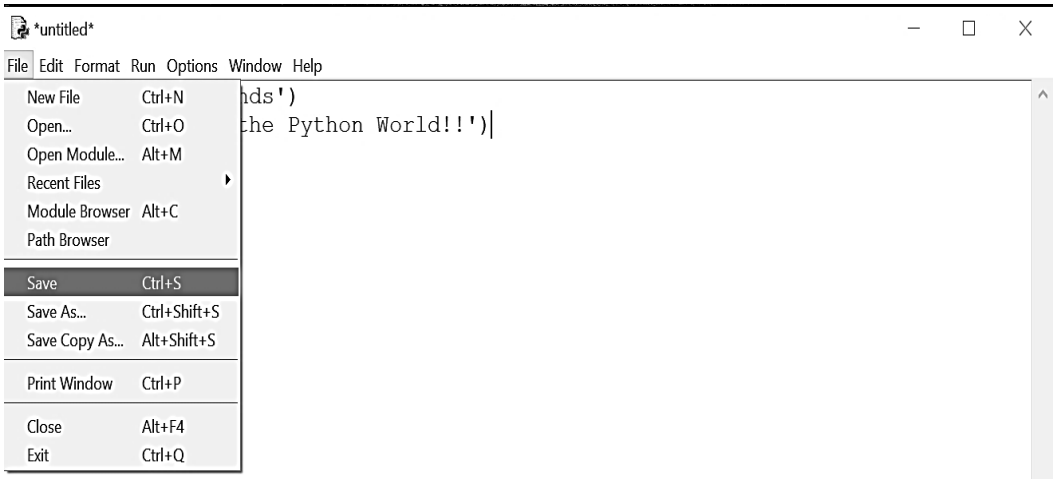


2. From File Menu Choose – New File

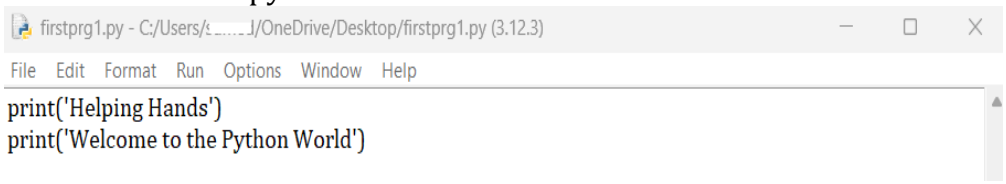


3. Type the codes on File named 'Untitled'

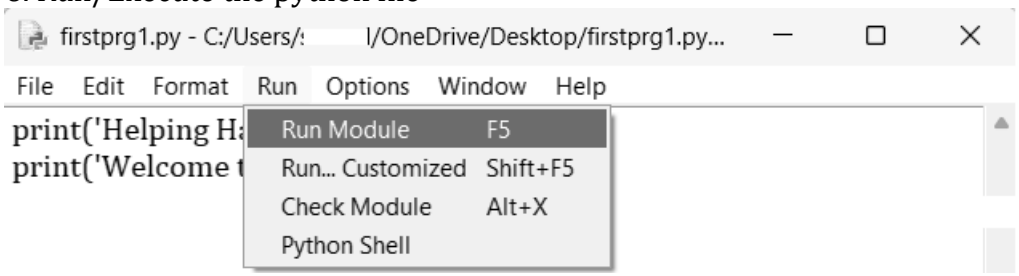
4. Click on Save from file menu



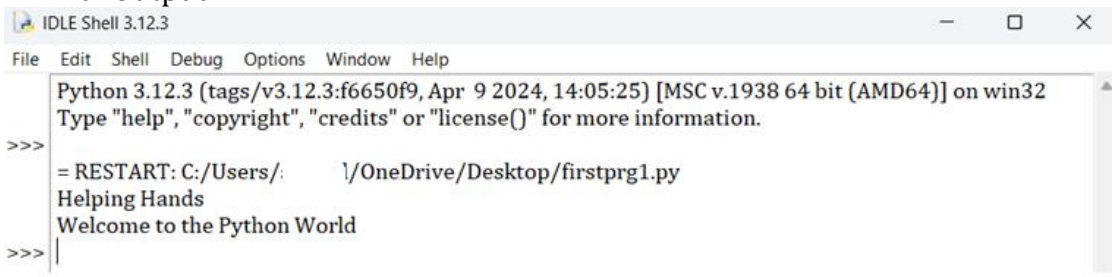
5. You will have a python file like this



6. Run/Execute the python file



7. Final Output



Interactive Mode Vs Script Mode

- Interactive Mode does not save the command entered by you in the form of a program but in Script mode program can be saved for later use.
- In Interactive Mode the output is sandwiched between in the command lines while in Script Mode the output of a program is displayed all together.
- In Interactive Mode, the output can be displayed as well as can be done using print function.
- But in Script mode, print ( ) command is preferably used to print results.

PYTHON CHARACTER SET

- Character set is a set of valid characters that a language can recognize
- It can be any letter, digit or any other special character. #, . ; &* () _/+=‘’ “” !
- It supports Unicode encoding standard
- ASCII – American Standard Code for Information Interchange
- Universal coding
- Letters from A-Z, a-z
- Digits 0-9
- Special Characters → space, +, - * / ** \ ( ) [ ] {} // = != == < > . ; : % ! & # <= >= @ _(underscore)
- White spaces → blank space, tabs (↹) , carriage return(↲), new line, form feed
- Other Characters → Python can process all ASCII and Unicode characters as part of data or literals.

PYTHON TOKENS

- Token is the smallest individual unit of program
- Also called as Lexical units or Lexical elements
- Types of tokens:
  - Keyword
  - Identifier
  - Literal

- Operator
- Punctuator

1.) Keywords – These are reserve words which has special meaning to the compiler or interpreter.

Python keywords:

FALSE	None	True	and	as	break	class	continue	def	pass
for	from	global	if	import	in	is	not	or	raise
del	elif	else	except	finally	with	return	try	while	

2.) Identifiers – These are names given to the different parts of the program like variables, objects, classes, functions, lists, dictionaries etc.

Identifier forming rules :

- Valid combination of letters and digits.
- First character can be a letter(A-Z ,a-z) or underscore(__) but not a digit(0-9).
- Upper case and lowercase alphabets are treated differently.
- Unlimited in length.
- Must not be a keyword eg break, print
- Cannot contain any special character except _ (underscore)

Valid Identifiers: Myfile, MYFILE, DATE_9, Z2TOZ9, _CHK, FILE13

Invalid Identifiers: DATA-REC, break, my.File, 19ct, First Name

3.) Literals - Literals are fixed or constant values used directly in code. They represent data you assign to variables or use in expressions.

Types of literals

- **String Literals →** Text enclosed in single (‘),double (") or triple(‘’’) quotes.

  Example -
  name = "Alice"
  greeting = 'Hello'
  message = ''' new things
  new ways'''

- **Numeric Literals →** Represent numbers of different types:

  Integer- Similar as integers in Maths ex. t=+7, y =-98
  Without decimal points
  Can have + or – sign (by default +)
  Commas cannot be used
  Example : 12, -17, 100000, 1220000 n1=-1234, n2 = +34

  Float - Numbers with decimal points, real numbers
  Can have + or – sign
  Example : 0.3 , 56.8 , -34.8, +90.1

  Complex – a+bj
  Example : 3+5j, -2j

- **Boolean Literals →** True(1), False(0)
- **Special Literal → None -** Represents the absence of a value or null.
  Example : data = None
- **Literal Collections → Literal collections** are ways to store multiple items in a single variable using built-in Python data structures. They include **lists**, **tuples**, **dictionaries**, and **sets**.

Type	Syntax Example	Ordered	Mutable	Duplicates Allowed
List Literal	["a", "b", "c"]	Yes	Yes	Yes
Tuple Literal	("x", "y", "z")	Yes	No	Yes
Dictionary Literal	{"key": "value"}	No	Yes	Keys: No, Values: Yes
Set Literal	{1, 2, 3}	No	Yes	No

4.) Operators

Operators in Python are special symbols that let *Python do something with values* (called operand).
Each operator type serves a different purpose.

◈ Some commonly used types of Python Operators:

- Arithmetic Operators – Perform basic math
  + - * / % // **
  *(Example: a + b, a * b)*
- Relational (Comparison) Operators – Compare values
  == != > < >= <=
  (Example: a > b, a == b)
- Logical Operators – Combine conditions
  and or not
  (Example: a > 5 and b < 10)
- Assignment Operators – Assign values
  = += -= *= /= etc.
  (Example: x += 1 is same as x = x + 1)

(*Topic Operator explained later*)

Punctuators - Punctuators are the symbols that are used in programming languages to organize programming sentence structures, and indicate the rhythm and emphasis of expressions, statements and program structure.
Most common punctuators of Python programming Language are:

‘ “ \ () [] {} @ , : . =

Variables - A variable in a program is uniquely identified by a name (identifier). Variable in Python refers to an ***object — an item or element that is stored in the memory***. Value of a variable can be used and processed during program run.
Example - num=89, marks = 35
Name = ‘Joseph’, c=2+3i

Dynamic typing - A variable pointing to a value of certain type, can be made to a point to a value/object of different type. This is called **DYNAMIC TYPING.**
Example

```
x=25
print(x)
x= ‘Hello’ # same variable x is now string
print(x)
```

Concept of L-value and R-value

L-value (Left Value) - A **L-value** refers to a location that **can appear on the left side** of an assignment.
It **must be a variable** or an object that can store a value.

Example:

```
x = 10
  # Here, `x` is a L-value because it refers to a memory location where 10 is stored.
```

R-value (Right Value)- A R-value is any expression or value that **can appear on the right side** of an assignment.
It **represents data (a value), not a location.**

Example:

```
x = 10
# Here, `10` is a R-value — it's the value being assigned to x.
```

COMMENTS - Comments are the lines or statements that are ignored by the compiler/ interpreter.
These are additional readable information to clarify the source code.
It enhances the readability of the program
It is of two types –

Single Line comment

This type of comment begins with a symbol # (Pound or hash) character and end with end of physical line.

Multiline Comment or Block Comment can be given in two ways:

1. Add a # symbol in the beginning of every physical line part of multi-line comments.
   Example

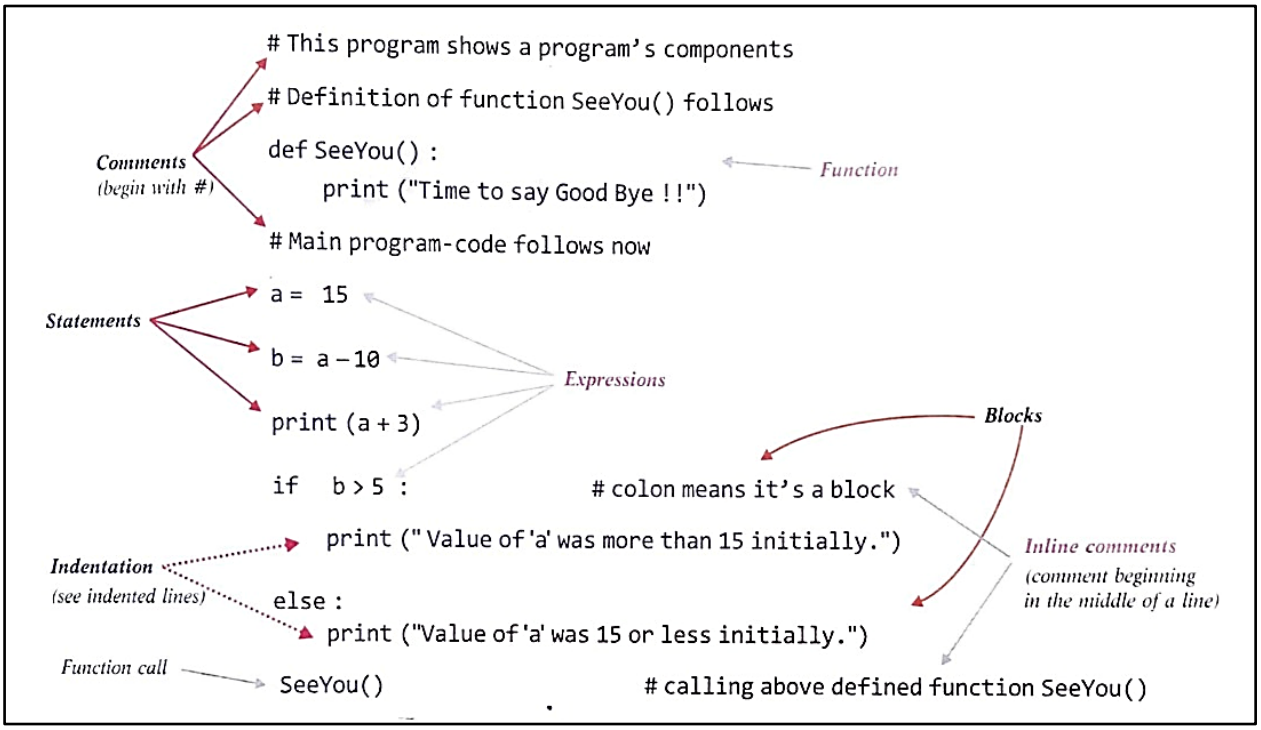
```
# Multiline comments are useful for detailed
# additional information related to program
#It helps clarify certain important things.
```

2. Type comment as a triple- quoted multi-line string
   Example

```
'''multiline comments are useful for detailed
additional information related to program.
It helps clarify certain important things.'''
```

<u>This type of multi-line comment is also called as *doc string* .</u>
The docstrings are very useful in documentation.

BAREBONES OF PYTHON



The above given sample program contains:

- Expressions
- Statements
- Comments
- Blocks and Indentations

EXPRESSION - An expression is any legal combination of symbols and values that represents a value.

Example

- (18/2) +15 – Arithmetic Expression
- (12>34) - Relational Expression
- a and b not b – Logical Expression
- 'and' + 'else' – String Expression

STATEMENT – A Statement is a programming instruction that does something ie some action take place.

Example
print('hello')
print(a+5)

BLOCKS AND INDENTATION - A group of statements which are part of another statement or a function are called block or code-block or suite in Python.

Example

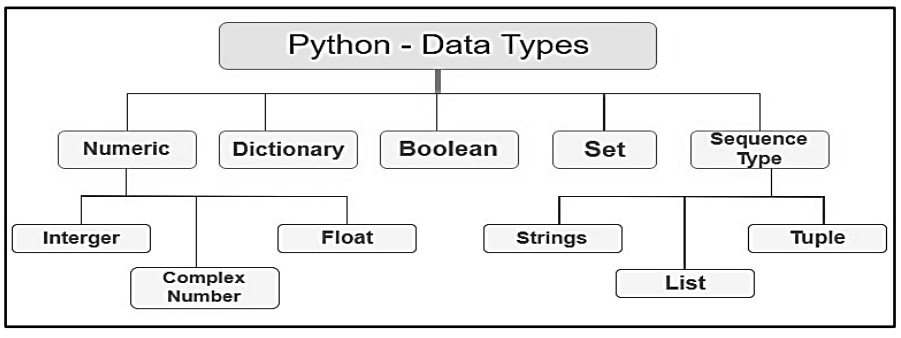
```
a= 2
b=90
if a>=23:
        print(" Eligible to vote")
        print("Have a nice day")
else:
        print("Not eligible")
print("Good Bye….")
```

PYTHON DATA TYPES

- Data Type specifies which type of value a variable can store.
- type ( ) function is used to determine a variable's type in Python.

Example

```
>>> x=98.75
>>> type(x)
<class 'float'>
```



NUMBER DATA TYPE –

1. **Integers** – This value is represented by int class. It contains positive or negative whole numbers (without fractions or decimals).
   Example : 12, -45

2. **Float** – This value is represented by the float class. It is a real number with a floating-point representation. It is specified by a decimal point.
   i)Fractional form or normal decimal notation:

Example : 3.14, 234.5678
ii)Exponential Notation
Example - 0.5E+5, 4.5E-10

3. **Complex Numbers** – A complex number is represented by a complex class. It is specified as *(real part) + (imaginary part)j* .
Complex Number is a number of the form A + Bj, where j is the imaginary number, **equal to the square root of -1 i.e.** $\sqrt{-1}$ **.**
Example - 5-6j real part=5, imaginary part -6

BOOLEAN DATA TYPE - Boolean values are the two constant objects False and True.

```
>>> a=2
>>> b=3
>>> c=a>=b
>>> print(c)
False
>>> print(type(c))
<class 'bool'>
```

SEQUENCE DATA TYPES IN PYTHON

Sequence data types store multiple values in an ordered way.
Types:

- String (str) – Text, e.g., "hello"
- List (list) – Ordered, changeable, e.g., [1, 2, 3]
- Tuple (tuple) – Ordered, unchangeable, e.g., (1, 2, 3)

MAPPING DATA TYPES IN PYTHON

Mapping data types store key-value pairs.

- Dictionary (dict) – Unordered collection of key: value pairs
  Example: {"name": "Ravi", "age": 12}

NONE DATA TYPE

None is a special data type in Python that represents **"nothing"** or **no value**.

```
x = None
print(x)  # Output: None
```

MUTABLE AND IMMUTABLE DATA TYPES

In Python, data types are divided into two categories:
Mutable: Data can be changed after it is created.
Immutable: Data cannot be changed once it is created.

How does Python decide?

When a value is created, Python stores it at a memory location.

- If Python allows the **same memory location to be updated**, it is **mutable**.
- If Python creates a **new object with a new memory address** on change, it is **immutable**.

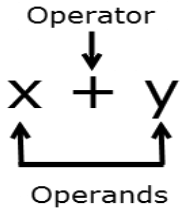
Data Type	Mutable / Immutable
int	Immutable
float	Immutable
str	Immutable
tuple	Immutable
bool	Immutable
list	Mutable
dict	Mutable
set	Mutable

Example to Understand:

IMMUTABLE	MUTABLE
a = "hello" print(id(a)) # say memory address is 101 a = a + " world" print(id(a)) # new address, e.g., 202	b = [1, 2, 3] print(id(b)) # say memory address is 303 b[0] = 100 print(id(b)) # still 303

OPERATORS

- The symbols that trigger the operation / action on data are called Operators and the data on which the operation is performed is called as Operands.
- Operators are the constructs which can manipulate the value of operands.
- Consider the expression 4 + 5 = 9. Here, 4 and 5 are called operands and + is called operator.
- The operand specifies the data that is to be manipulated by the operator.



TYPES OF OPERATORS

Python language supports the following types of operators:

- Arithmetic Operators
- Comparison (Relational) Operators
- Assignment Operators
- Logical Operators
- Membership Operators
- Identity Operators

Arithmetic Operators - Used to perform mathematical operations like addition, subtraction, multiplication and division etc.

OPERATOR	DESCRIPTION	EXAMPLE
+ Addition	Adds values on either side of the operator.	a=10, b=20 a + b = 30
- Subtraction	Subtracts right hand operand from left hand operand.	a – b = -10
* Multiplication	Multiplies values on either side of the operator	a * b = 200
/ Division	Divides left hand operand by right hand operand	b / a = 2
% Modulus	Divides left hand operand by right hand operand and returns remainder	b % a = 0
** Exponent	Performs exponential (power) calculation on operators	a**2 =100
// Floor Division	The division of operands where the result is the quotient in which the digits after the decimal point are removed. But if one of the operands is negative, the result is floored, i.e., rounded away from zero (towards negative infinity) –	9//2 = 4 , 9.0//2.0 = 4.0, -11//3 = -4, -11.0//3 = -4.0

Comparison (Relational) Operators - Relational operators compares the values. It either returns True or False according to the given condition.

Operator	Description	Example
==	If the values of two operands are equal, then the condition becomes true.	(a == b) False
!=	If values of two operands are not equal, then condition becomes true.	(a != b) True 7!=23 True
>	If the value of left operand is greater than the value of right operand, then condition becomes true.	(a > b) False
<	If the value of left operand is less than the value of right operand, then condition becomes true.	(a < b) True
>=	If the value of left operand is greater than or equal to the value of right operand, then condition becomes true.	(a >= b) False 7>= 98 False
<=	If the value of left operand is less than or equal to the value of right operand, then condition becomes true.	(a <= b) True 8<=9

Assignment Operators - The **assignment operator** is used to **assign values to variables** in Python.

Operator	Description	Example
=	Assigns values from right side operands to left side operand	c = a + b assigns value of a + b into c , j=78
+= Add AND	It adds right operand to the left operand and assign the result to left operand	c += a is equivalent to c = c + a
-= Subtract AND	It subtracts right operand from the left operand and assign the result to left operand	c -= a is equivalent to c = c - a
*= Multiply AND	It multiplies right operand with the left operand and assign the result to left operand	c *= a is equivalent to c = c * a
/= Divide AND	It divides left operand with the right operand and assign the result to left operand	c /= a is equivalent to c = c / ac /= a is equivalent to c = c / a
%= Modulus AND	It takes modulus using two operands and assign the result to left operand	c %= a is equivalent to c = c % a
**= Exponent AND	Performs exponential (power) calculation on operators and assign value to the left operand	c **= a is equivalent to c = c ** a
//= Floor Division	It performs floor division on operators and assign value to the left operand	c //= a is equivalent to c = c // a

Logical Operators - Logical operators perform Logical AND, Logical OR and Logical NOT operations. It also returns either True or False.

OPERATOR	DESCRIPTION	Example
and	Logical AND: True if both the operands are true otherwise False	x and y
or	Logical OR: True if either of the operands is true otherwise False	x or y
not	Logical NOT: True if operand is False and vice versa	not x

Membership Operators - Python's membership operators test for membership in a sequence, such as strings, lists, or tuples. This also results in True or False. There are two membership operators as explained below –

str= 'HELLO 123#'

'X' in str

Operator	Description	Example
in	Evaluates to true if it finds a variable in the specified sequence and false otherwise.	x in y, here in results in a 1 if x is a member of sequence y.
not in	Evaluates to true if it does not finds a variable in the specified sequence and false otherwise.	x not in y, here not in results in a 1 if x is not a member of sequence y.

Identity Operators - Identity operators compare the memory locations of two objects.

is and is not are the identity operators both are used to check if two values are located on the same part of the memory. Two variables that are equal does not imply that they are identical.

Operator	Description	Example
is	Evaluates to true if the variables on either side of the operator point to the same object and false otherwise.	x is y, here is results in 1 if id(x) equals id(y).
is not	Evaluates to false if the variables on either side of the operator point to the same object and true otherwise.	x is not y, here is not results in 1 if id(x) is not equal to id(y).

PRECEDENCE OF OPERATORS

Evaluation of the expression is based on precedence of operators. When an expression contains different kinds of operators, precedence determines which operator should be applied first. Higher precedence operator is evaluated before the lower precedence operator

Sr.No.	Operator	Description
1	** 2**3	Exponentiation (raise to the power)
2	~ + -	Complement, unary plus and minus (method names for the last two are +@ and -@)
3	* / % //	Multiply, divide, modulo and floor division
4	+ -	Addition and subtraction
5	>> <<	Right and left bitwise shift
6	&	Bitwise 'AND'
7	^ |	Bitwise exclusive `OR' and regular `OR'
8	<= < > >=	Comparison operators
9	<> == !=	Equality operators
10	= %= /= //= -= += *= **=	Assignment operators
11	is, is not	Identity operators
12	in,not in	Membership operators
13	not, or, and	Logical operators

Example : 2*3**2 this will result in 18 as exponent has higher precedence than multiply.

TYPE CONVERSION

Type conversion means we can change the data type of a variable in Python from one type to another.

It is categorised into two types:

1. Implicit Type Conversion (Automatic)

Python automatically converts data type during operations when safe.

```
a = 5     # int
b = 2.0   # float
c = a + b  # Result: 7.0 (float)
```

2. Explicit Type Conversion (also called Type Casting)

You manually convert data using functions like:

- int() → Converts to integer
- float() → Converts to float
- str() → Converts to string
- list(), tuple(), etc.

```
x = "123"
y = int(x)  # Now y is an integer 123
```

More Examples:

int(3.3) produces 3
float(3) produces 3.0
str(3.3) produces "3.3"
float("3.5") produces 3.5

SIMPLE INPUT AND OUTPUT

- To get input from user use input()
- By default input function

<variable> = input ("<Prompt to be displayed>")

This will pause the program and wait for the user to type something.

Note : Whatever the user types is taken as a string.

Example

```
name = input ('Enter your name') #taking text input
age = (input ('Enter your age'))    # although user wants a number but input takes
                                    #string by default.
```

Reading Numbers

```
age = int (input ('Enter your age')) #input string converted into integer
                                     #suppose user entered 10
g = age+12                           #10+12
print(g)                             #output - 22
```

Output through print () – display the result

print(object, [sep =' ' or <separator string > end = '\n' or <end-string>])

Example:

```
print("Hello", "World", "!!!!", sep="-")
output: Hello-World-!!!!

print("Hello", end=" ")
print("World")
output: Hello World
```

DEBUGGING

- Debugging refers to the process of locating the place of error, cause of error and removing the errors by rectifying the code accordingly.

- An error in program caused while executing it or producing incorrect output is called as ***Program bug.***
- An error called as bug is the code that prevents a program from compiling and running correctly OR DESIRED OUTPUT.
- These are of three types:
  - **Compile Time Error**
  - **Run Time Error**
  - **Logical Error**

1. COMPILE TIME ERROR - The errors that occurs during compile – time are called as *Compile time errors.*

When a program is compiled, it checks whether it follows the programming languages rules or not.

It is of two Types:

- Syntax Errors
- Semantics Error

Syntax Errors-

- Syntax refers to formal rules governing the construction of valid statements in a language.
- Python's indentation errors (wrong indentation) are syntax errors.

Example –

```
X =9
X=<23  #  x <=23
PRINT(x) # print(x)
if a<0:
        print 'negative'
```

Semantics Error-

- "Semantics refers to the set of rules which give the meaning of a statement."
- This error occurs when the statement is not meaningful.

Example –

```
 x*y =z
Z=X*Y
An expression cannot be on left side of assignment statement.
a>=q if :
If a>=q:
```

2. LOGICAL ERROR –

- Its executes the program successfully but you will not get desired output
- For example – An incorrectly implemented algorithm or use of variable before initialization, wrong end of loop etc.
- It is often hardest to prevent and locate.

Example - To Multiply 3 nos.

```
a= 9
b =12
c=a*b
print (c)
```

3. RUN TIME ERROR –

A **run-time error** is an error that occurs **while the program is running**, causing it to **stop or crash**. Such error happens **after the program starts**, when Python **cannot complete an instruction** due to a problem during execution.

Examples:

- Dividing by zero → 10 / 0 (ZeroDivisionError)

- Using a value that doesn't exist → print(x) (when x is not defined)
- Accessing an invalid index → list[10] (if the list is smaller)

QUESTION BANK

	MULTIPLE CHOICE QUESTIONS
1.	What is the first phase in the Python project cycle? a) Design b) Development c) Requirement Analysis d) Testing
2.	What activity is commonly done in both the testing and maintenance phases? a) Writing user stories b) Fixing bugs c) Creating diagrams d) Writing documentation
3.	Which of the following best defines an algorithm? a) A diagram showing steps b) A step-by-step procedure for solving a problem c) A computer program d) A software model
4.	What is a flowchart used for? a) Drawing architectural models b) Writing code c) Visually representing algorithms d) Designing databases
5.	In a flowchart, which symbol is used to represent a decision? a) Rectangle b) Parallelogram c) Oval d) Diamond
6.	Who developed Python? a) Dennis Ritchie b) Guido van Rossum c) Bjarne Stroustrup d) James Gosling
7.	Which of the following is used to output data in Python? a) echo() b) write() c) print() d) show()
8.	What is the correct way to create a variable in Python? a) var x = 10 b) int x = 10 c) x := 10 d) x = 10
9.	What data type is the result of: type("123")? a) int b) str c) float d) bool
10.	Which of the following is NOT a Python data type? a) list b) tuple c) array d) dictionary
11.	Python is: a) Compiled b) Interpreted c) Both d) None
12.	Which of these is not a core data type? a) List b) Dictionary c)Tuple d) Class
13.	What is the output of this code? x = None print(type(x)) a) <class 'null'> b) <class 'empty'> c) <class 'NoneType'> d) <class 'str'>
14.	Which of the following is a mutable data type? a) list b) tuple c) str d) int
15.	What will be the result of 3 + 2 * 2? a) 10 b) 7 c) 9 d) 8
16	What is the result of bool (0) in Python? a) True b) False c) 0 d) Error
17.	What type of error is raised by the following code?

	x = 10 print(x / 0) a) NameError b) TypeError c) ValueError d) ZeroDivisionError
18.	Which of the following is a valid Python variable name? a) 1variable b) @name c) _value d) class
19.	What will this code output? a = [1, 2, 3] print(a[3]) a) 3 b) IndexError c) None d) 4
20.	Which of the following is NOT a keyword in Python? a) None b) pass c) eval d) finally

ANSWERS

1	2	3	4	5	6	7	8	9	10
c	b	b	c	d	b	c	d	b	c
11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
b	d	c	a	b	b	d	c	b	c

ASSERTION AND REASONING QUESTIONS

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

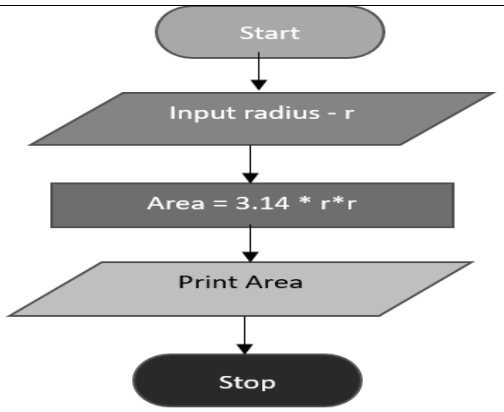
A. Both A and R are true, and R is the correct explanation of A.
B. Both A and R are true, but R is not the correct explanation of A.
C. A is true, but R is false.
D. A is false, but R is true.

1.	**Assertion (A):** int and float are both immutable data types. **Reason (R):** Reassigning a number to a variable changes the data inside the same memory location.
2.	**Assertion (A):** The None keyword in Python represents the absence of a value. **Reason (R):** None can be used in conditional statements and evaluates to False.
3.	**Assertion (A):** Lists in Python are mutable. **Reason (R):** Lists do not support indexing and slicing.
4.	**Assertion (A):** Division by zero in Python raises an error. **Reason (R):** The interpreter cannot represent infinity.
5.	**Assertion (A):** The expression 4 + 5 * 2 evaluates to 18. **Reason (R):** In Python, operators are evaluated from left to right without any precedence.
6.	**Assertion (A): is** operator checks if two variables refer the same object in memory. **Reason (R):** The == operator compares values, not identities.
7.	**Assertion (A):** x += 3 is a shorthand for x = x + 3. **Reason (R):** += is a compound assignment operator that modifies the variable in-place for immutable types.
8.	**Assertion (A):** The expression 10 // 3 results in 3. **Reason (R):** // operator performs floor division in Python.

ANSWERS

1.	Answer: C Yes, they are immutable, but reassigning creates a new object, not changes the value in-place.
2.	Answer: A

3.	Answer: C *Lists are mutable, but they* **do support** *indexing and slicing.*
4.	Answer: B Python raises a ZeroDivisionError for mathematical correctness, not because it can't represent infinity (it can, e.g., via float('inf')).
5.	Answer: C Correct result is 14. Operator precedence means * is evaluated before +.
6.	Answer: A
7.	Answer: C x += 3 does not modify in-place if x is immutable (like an int) — it creates a new object.
8.	Answer: A
SHORT ANSWER QUESTIONS	
1.	Explain Problem Solving and give its steps?
2.	Write an algorithm to find the largest of three numbers.
3.	Draw a flowchart to calculate area of circle
4.	Differentiate between Interactive mode and Script mode?
5.	Differentiate between Keyword and Identifiers?
6.	What are Literals in Python? Identify the following type of literals- a) x='Rehan' b) y=3+6j c) t= 'True' d) p=3.15
7.	What is the purpose of comments in Python? In how many types we can give comments in Python explain it.
8.	Differentiate between mutable and immutable data types?
9.	Differentiate between List and Dictionary with example?
10.	When Tuple is preferred over List?
11.	What is type casting? Explain with example.
12.	Differentiate between Syntax Error and Run time error?
	ANSWERS
1	Problem Solving is the process of identifying a problem, analyzing it, and finding an effective solution, often by writing a program. It is a key skill in programming and computational thinking. **Steps of Problem Solving** 1. Analysing the problem, 2. Developing an algorithm, 3. Coding, 4. Testing, and debugging
2	**Step 1:** Start **Step 2:** Input three numbers: A, B, and C **Step 3:** Check If A > B and A > C, then print A is the largest **Step 4:** Else if B > C, then Print B is the largest

	Step 5: Else Print C is the largest **Step 6:** Display the largest number **Step 7:** Stop
3	
4	1. Interactive Mode does not save the command entered by you in the form of a program but in Script mode program can be saved for later use. 2. In Interactive Mode the output is sandwiched between in the command lines while in Script Mode the output of a program is displayed all together. 3. In Interactive Mode, the output can be displayed as well as can be done using print function. But in Script mode, print() command is preferably used to print results.
5	**Keywords** are reserve words which has special meaning to the compiler or interpreter. Python keywords: False, None, True, and , as, break etc **Identifiers** are names given to the different parts of the program like variables, objects, classes, functions, lists, dictionaries etc. Example - age, student_name, totalSum, calculate_area
6	Literals are fixed or constant values values used directly in code. They represent data you assign to variables or use in expressions a) x='Rehan' - String Literal b) y=3+6j – Complex Literal c) t= 'True' – String Literal d) p=3.15 - Float literal
7	Comments are the lines or statements that are ignored by the compiler/ interpreter. These are additional readable information to clarify the source code. It enhances the readability of the program It is of two types – **Single Line comment** - This type of comment begin with a symbol # (Pound or hash)character and end with end of physical line. **Multiline Comment or Block Comment** can be given in two ways: 1. Add a # symbol in the beginning of every physical line part of multi-line comments. 2. Type comment as a triple- quoted multi-line string
8	**Immutable data type** - **Cannot change** once created (any change creates a new object). These data types cannot be changed after they are created.

	<table><tr><th>Type</th><th>Example</th></tr><tr><td>int</td><td>x = 10</td></tr><tr><td>float</td><td>pi = 3.14</td></tr><tr><td>bool</td><td>flag = True</td></tr><tr><td>str</td><td>Name="Alice"</td></tr><tr><td>tuple</td><td>colors = ("red",”green”,”blue”)</td></tr></table> **Mutable Data types Can change** after creation (modify content without changing identity). <table><tr><th>Type</th><th>Example</th></tr><tr><td>list</td><td>nums = [1, 2, 3]</td></tr><tr><td>dict</td><td>info = {"name": "Bob",’Gender’:’Male’,’Age’:23}</td></tr></table>
9	<table><tr><th>Feature</th><th>List</th><th>Dictionary</th></tr><tr><td>Structure</td><td>Ordered sequence</td><td>Unordered key-value pairs</td></tr><tr><td>Access</td><td>By index (e.g., list[0])</td><td>By key (e.g., dict["key"])</td></tr><tr><td>Syntax</td><td>Square brackets []</td><td>Curly braces {}</td></tr><tr><td>Key-Value</td><td>No (only values)</td><td>Yes</td></tr><tr><td>Example</td><td>["a", "b", "c"]</td><td>{"a": 1, "b": 2}</td></tr></table>
10	**Tuples are immutable**, meaning their values **cannot be changed** after creation. This makes them **more secure and reliable** when storing **constant data.**
11	**Type casting** is the process of **converting one data type into another** manually. Python supports two types of casting: 1. **Implicit Type Casting** (done automatically) 2. **Explicit Type Casting** (done by the programmer) Examples: int(3.3) produces 3 str(3.3) produces “3.3”

12	**Syntax Error -** Occurs when the rules of the programming language are broken.
	Detected before the program runs (during compilation).
	Example –
	X =9
	X=<23 # x <=23
	PRINT(x) # print(x)
	if a<0:
	print ‘negative’
	Run-time Error - Occurs while the program is running**.**
	Syntax is correct, but something goes wrong during execution
	Example - dividing by zero, accessing invalid index .

CASE STUDY BASED QUESTIONS

1. ABC school wants to develop a simple Python script to store a student's name, roll number, and marks in three subjects. They also want to calculate the average marks and check if the student has passed (pass mark is 33 in each subject).

Questions:

1. Identify the appropriate data types for:
   - Student name
   - Roll number
   - Marks
2. Write a Python code to store the above data and calculate average marks.
3. Use operators to check if the student passed in all subjects.
4. What error will occur if a string is added to an integer in the marks calculation? Show how to correct it.

2. Rehan wants to create an employee salary calculator which takes basic salary and calculates HRA (20%) and DA (50%), then calculates the gross salary.
Gross Salary = Basic Salary+HRA+DA

Questions:

1. Define variables to store basic salary, HRA, and DA.
2. Write the code to calculate gross salary.
3. Which operator will be used if we don't want decimal values?
4. Identify and fix the error if someone mistakenly writes
   HRA = basic_salary * 20/100%.

ANSWERS

1

1. Student name → str
   Roll number → int or str (depending on format)
   Marks → int or float
2.
```
name = "Alice"
roll number = 101
mark1 = 78
mark2 = 65
mark3 = 80
average = (mark1 + mark2 + mark3) / 3
print ("Average Marks:", average)
```
3.
```
if mark1 >= 33 and mark2 >= 33 and mark3 >= 33:
  print("Passed")
else:
  print("Failed")
```
4. TypeError: can only concatenate str (not "int") to str

2

1.
```
basic_salary = 20000
hra = basic_salary * 0.20
da = basic_salary * 0.50
```
2.
```
gross_salary = basic_salary + hra + da
print ("Gross Salary:", gross_salary)
```
3. // floor division
4. SyntaxError: invalid syntax due to % symbol in 20/100%
```
hra = basic_salary * 20 / 100  # or basic_salary * 0.20
```

COMPETANCY BASED QUESTIONS

1

Question:
You are building a banking system where users can deposit and withdraw money.

	a.) Declare variables for account holder name, balance, and last transaction amount. b.) Write code to update the balance after a deposit of ₹1500 and a withdrawal of ₹800.
2	**Question:** An e-commerce platform calculates discounts based on the purchase amount. If the amount is more than ₹5000, a 10% discount applies. Write a Python program to take an amount as input and display the final price after applying the discount if applicable.
3	**Question:** You are developing a health tracking app. Each user logs their name, age, weight (in kg), height(in m) and whether they exercised today (Yes/No). a.) Identify appropriate data types for each of these attributes. b.) Write a program to calculate BMI(weight/height^2) according to values entered.
ANSWERS	

1	**Competency Focus:** Declare and use variables for storing and updating data.
	a.)
	account_holder = 'Neha'
	balance = 10000
	last_transaction = 0
	
	b.)
	# Deposit
	last_transaction = 1500
	balance += last_transaction
	# Withdrawal
	last_transaction = -800
	balance += last_transaction
	print ('Updated Balance:', balance)

2	**Competency Focus:** Use arithmetic, comparison, and logical operators appropriately.
	amount = float (input ("Enter purchase amount: "))
	if amount > 5000:
	  discount = amount * 0.10
	else:
	  discount = 0
	final_price = amount - discount
	print ("Final Price:", final_price)

3	**Competency Focus:** Select and use appropriate data types.
	a)
	- Name → `str`
	- Age → `int`
	- Weight → `float`
	-Height→'float'
	- Exercised → `bool` or `str`
	b)
	#input weight and height
	bmi = Weight/(Height*Height)
	print(bmi)

FLOW OF CONTROL

•Flow of Control: introduction, use of indentation, sequential flow, conditional and iterative flow •Conditional statements: if, if-else, if-elif-else, flowcharts
•Iterative Statement: for loop, range (), while loop, flowcharts, break and continue statements, nested loops

Introduction to Flow of Control

In a program, instructions usually run one after another, step by step. But in real life, we often need to make choices or do things more than once. **Control structures** help us do that in programming. They let us decide what to do next or repeat certain steps, making our programs smarter and more useful

Use of Indentation

Indentation shows which lines of code belong together, especially inside loops or if statements. In Python, it's very important! If the indentation is wrong, your code might not run, or it might do the wrong thing.

Three fundamental types of flow control in programming:

- Sequential Flow
- Conditional Flow
- Iterative Flow

Sequential Flow:

Sequential flow is the simplest form of flow control, where statements are executed one after another in the order they appear. This is the default mode of execution in most programming languages.

Example:

```
print("start")
print("process")
print("End")
```

Conditional Flow:

Conditional flow allows the program to make decisions and execute certain blocks of code based on specified conditions. In Python we use conditional statements like if, if-else, and if-elif-else.

Example:

```
age=int(input("enter your age pls"))
if age >= 18:
    print("You are eligible to vote.")
else:
    print("You are not eligible to vote.")
```

Output:

```
enter your age pls21
You are eligible to vote.
```

Iterative Flow:

Iterative structures enable the repetition of a block of code multiple times. In python we use **for** and **while** loops. These are essential for tasks that require repetition till certain no. of times (FOR) or until a condition is true. (WHILE)

Example:

```
for i in range(5):
    print("value of =",i)
```

Output:

```
value of = 0
value of = 1
value of = 2
value of = 3
value of = 4
```

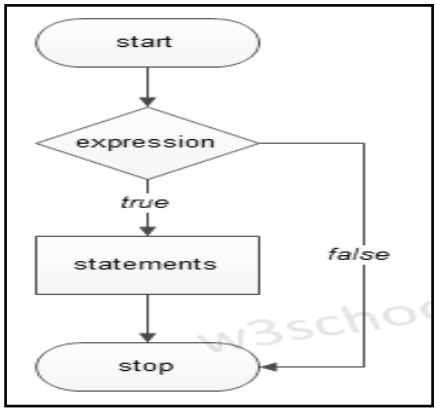
CONDITIONAL STATEMENTS

- if Statement
- if-else Statement
- if-elif-else Statement

if statement:

It is the simplest form of conditional statement. It evaluates a condition and executes a block of code only if the condition is true.

Syntax:




```
if condition:
    # code to execute if condition is true
```

Example:

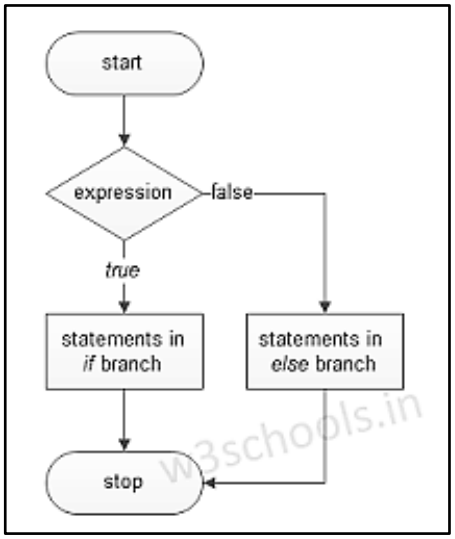
```
number=8
if number%2==0:
   print("number is even")
```

Output:

```
number is even
```

if-else Statement:

The if-else statement extends the if statement by providing an alternative block of code that executes when the condition is false.




```
if condition:
        # code to execute if condition is true
else:
        # code to execute if condition is false
```

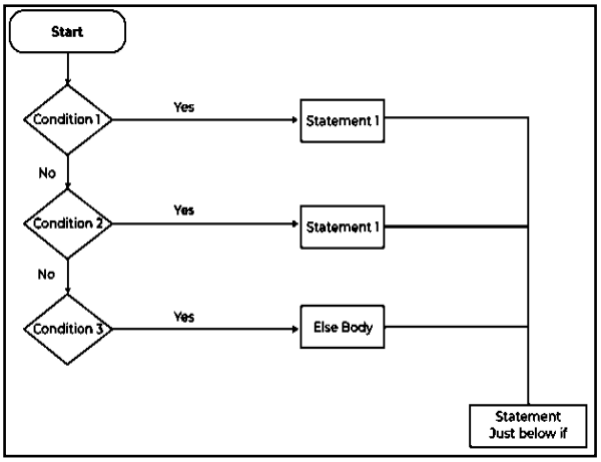
```
number=7
if number%2==0:
   print("number is even")
else:
   print("number is odd")
```

Output:

```
number is odd
```

if-elif-else Statement: The if-elif-else statement allows for multiple conditions to be checked sequentially. If the first condition is false, the program checks the next condition, and so on. If none of the conditions are true, the else block is executed.

```
if condition1:
        # code to execute if condition1 is true
elif condition2:
        # code to execute if condition2 is true
else:
```



Example:

```
number=7
if number%2==0:
   print("The number is even")
elif number%5==0:
   print("The number is divisble by 5")
elif number%7==0:
   print("The number is divisble by 7")
else:
   print("again check the number")
```

Output:

```
The number is divisble by 7
```

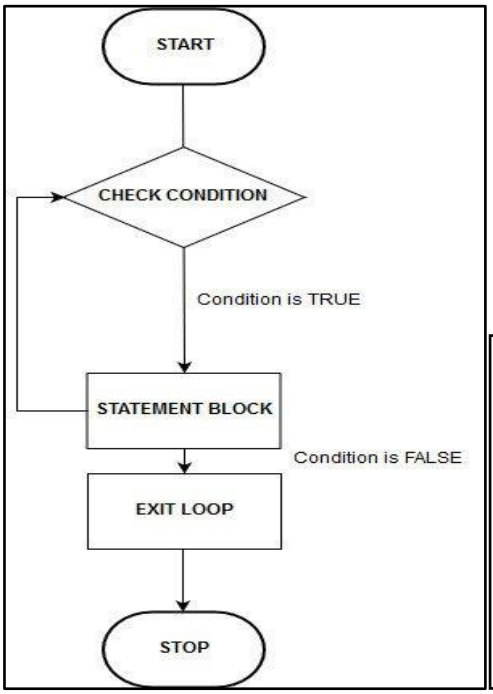
ITERATIVE STATEMENT

An iterative statement is used to repeat a block of code multiple times. Iterative statements help loop through data or perform tasks repeatedly based on conditions. There are two main types of iterative statements in Python:

1. for loop
2. while loop

for Loop:

for loops are used when we have a block of code (*A block in Python is a group of lines of code that are meant to run together*.) which we want to repeat a fixed number of times for each item in a sequence (like a list, string, or range of numbers etc.)




```
for item in seq:
    # block of code (loop body)
else: <optional>
    # This p art runs only when loop
    runs normally
```

Example:

```
seq = [1,2,3,4]
for i in seq:
  print(i)
else:
  print('loop completed normally')
```

Output:

```
1
2
3
4
loop completed normally
```

range() function in python:

The range( ) function in Python is used to **generate a sequence of numbers**, which is commonly used in for loops.

Syntax :

```
range(stop)              # starts from 0, ends at stop -1
range(start, stop)       # starts from 'start', ends at stop - 1
range(start, stop, step) # step controls increment (or decrement)
```

Example 1:

```
for i in range(5):
   print(i)
```

Output:

```
0
1
2
3
4
```

Example 2:

```
for i in range(2,6):
   print(i)
```

Output:

```
2
3
4
5
```

Example 3:

```
for i in range(1,10,2):
   print(i)
```

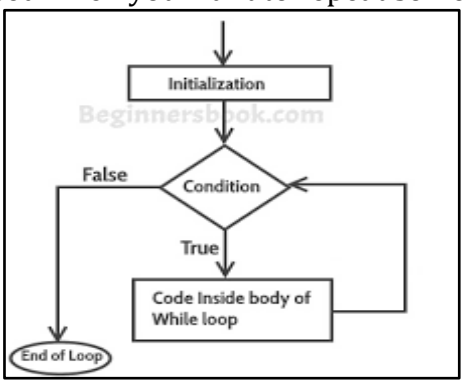
Output:

```
1
3
5
7
9
```

Example 4:	Output:
for i in range(10,0,-2): print(i)	10 8 6 4 2

while Loop:

Used when you want to repeat something as long as a certain condition is True.




```
while condition:
    # block of code (loop body)
```

Example:

```
num=0
while num<5:
  print("num is :",num)
  num+=1
```

Output:

```
num is : 0
num is : 1
num is : 2
num is : 3
num is : 4
```

Jump Statement

A **jump statement** is used to **change the normal sequential flow** of a program. Instead of executing code line-by-line in order, a jump statement "jumps" the execution to a different part of the code — either to **exit**, **skip**, or **return**.

1. **break**
2. **continue**

break Statement:

Immediately **exits** the loop, even if the condition is still true/iterations are left.

Example:

```
for i in range(10):
  if i==5:
    break
  print(i)
```

#loop stops when i becomes 5

Output:

```
0
1
2
3
4
```

continue Statement:

Skips the rest of the current loop iteration and moves to the next one.

Example:

```
for i in range(5):
  if i==2:
    continue
  print(i)
```

Output:

```
0
1
3
4
```

when i == 2, continue skips print(i) for that iteration.

Nested loop:

We call a loop, nested loop when one loop is inside another loop. This is often used for working with grids, tables, or multi-dimensional data (like matrices).

```
for outer in range(...):
    for inner in range(...):
```

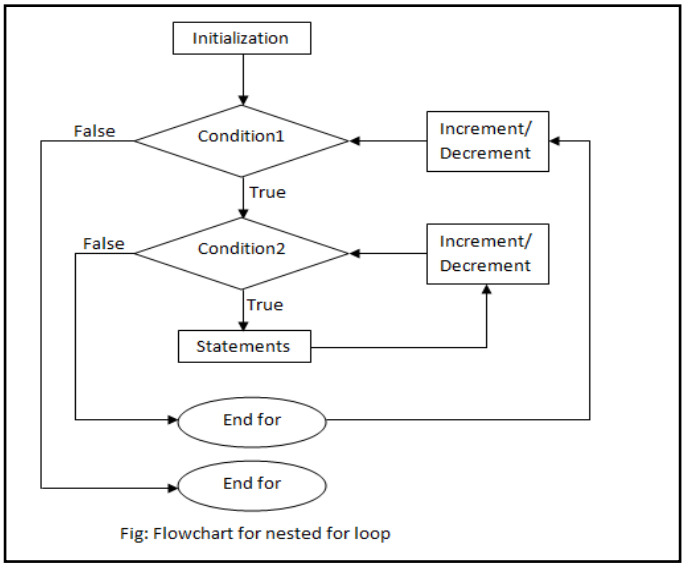


Fig: Flowchart for nested for loop

Example

```
for i in range (1,5):
  for j in range(i):
    print("*",end=" ")
  print()
```

Output

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

```
#Generating patterns
j=int(input("enter no of row"))
for i in range(1,j+1):
  for j in range(i):
    print("@",end= " ")
  print()
```

```
@
@ @
@ @ @
@ @ @ @
@ @ @ @ @
```

QUESTION BANK

MULTIPLE CHOICE QUESTIONS

1	What will be the output of the following code?
	x=-10
	if x>0:
	print("positive")
	else:
	print("non-positive")
	a) Positive b)Non-positive c)Negative d)Error

2	Which statement checks if a number n is divisible by 3 and 5? a) if n / 3 and n / 5: b) if n % 3 == 0 or n % 5 == 0: c) if n % 3 == 0 and n % 5 == 0: d) if n == 3 and 5:
3	Which loop is best when the number of iterations is unknown? a) for b) while c)do-while d)if
4	What is the output of this code?

	n=4 fact=1 for i in range(1,n+1): fact*=i print(fact) a) 10 b) 24 c)16 d) 5
5	Choose the correct flow for if-elif-else: a) Always executes all blocks b) Executes only one block c)May skip the if block d) Executes in reverse order
6	Which of the following will print the numbers from 10 to 1? a) for i in range(10): print(i) b) for i in range(1, 11): print(i) c) for i in range(10, 0, -1): print(i) d) for i in range(0, 10, -1): print(i)

7	What will be the output of this code?
	i=0
	while i<3:
	  print(i,end=" ")
	  i+=1
	a) 0 1 2 b) 1 2 3 c) 0 1 2 3 d)1 2

8	Which keyword is used to exit a loop early? a) continue b) exit c) break d)stop
9	Choose the correct statement about loops in Python: a) Only for loops can be nested b) break can only be used in for loops c) while and for loops can both use break and continue d) Python does not support nested loops
10	What will this print? for i in range(3): print("Hello",i,end="") a) Hello 1, Hello 2, Hello 3 b)Hello 0 Hello 1 Hello 2 c) Hello 0 Hello 2 Hello 4 d)Error

11	What is the output of this nested loop?
	for i in range(2):
	  for j in range(3):
	  print(i,j,end="")
	a) (0,0), (1,0), (2,0) b) 0 0 0 1 0 2 1 0 1 1 1 2 c) 2 times d)3 times

12	Which of the following is a valid **if** condition? a) if (x == 10) b) if x = 10 c) if x <== 10 d) if x => 10
13	What does range(5, 1, -1) generate? a) 5 4 3 2 b) 5 4 3 2 1 c) 5 3 1 d) 5 1

14	What is the output of this code?
	for i in range(2,6):
	  if i==4:
	  break
	  print(i, end="")
	
	a) 2 3 b) 2 3 4 5 c) 2 3 4 d) 2 3 5

15	Which of the following is NOT a valid loop in Python. a) for b) while c) do-while d) All are valid

ANSWERS

1. b) Non-positive	2. c) if n % 3 == 0 and n % 5 == 0:
3. b) while	4. b) 24
5. b) Executes only one block	6. c) for i in range (10, 0, -1): print(i)
7. a) 0 12	8. c) break
9. c) while and for loops can both use break and continue	10. b) Hello 0 Hello 1 Hello 2
11. b) 0 0 0 1 0 2 1 01 1 1 2	12. a) if (x == 10)
13. a) 5 4 3 2	14. a) 2 3
15. c) do while	

ASSERTION AND REASONING QUESTIONS

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

1. Both A and R are true and R is the correct explanation of A.
2. Both A and R are true but R is not the correct explanation of A.
3. A is true but R is false.
4. A is false but R is true.

Q,1. Assertion (A): break and continue are termed as Jump statements.
Reasoning (R): Jump statements can only be used with looping constructs but not with conditional constructs.

Q2. Assertion (A): In an if-else statement, the if block is executed if condition is TRUE whereas else block is executed if condition is FALSE.
Reasoning (R): In a conditional construct, else block is mandatory.

Q3. Assertion (A): The continue statement skips the rest of the current loop iteration and moves to the next one.
Reason (R): The continue statement ends the entire loop immediately.

Q4. Assertion (A): An elif block can exist without an else block.
Reason (R): elif provides additional conditional checks after the initial if.

Q5. Assertion (A): range(5) generates numbers from 0 to 4.
Reason (R): The range( ) function includes the start value but excludes the end value.

ANSWERS

1. (A) Both A and R are true and R is the correct explanation of A.
2. (C) A is true but R is false.
3. (C) A is true but R is false.
4. (B) Both A and R are true but R is not the correct explanation of A
5. (A) Both A and R are true and R is the correct explanation of A.

SHORT ANSWER QUESTIONS

Q1. What is the difference between else and elif constructs. .

Q2. Convert the following for loop code into while loop.

```
for x in range (1,100,2):
print(x)
```

Q3 .What is an infinite while loop? Give an example.

Q4. Write a for loop that prints numbers from 10 down to 1.

Q5. Write a range( ) statement that generates numbers from 1 to 20 (including both)

Q6. What does the break and continue statement do in a loop?

Q7. Which keyword is used to add multiple dependent conditions in a statement? Write a small code to explain.

Q8 . What will this code print?

```
i=10
while i<8:
  print(i)
  i-=1
```

Q9. Find output :

```
for i in range(3):
  if i==1:
    break
  print(i)
```

Q 10. Write a program to accept a word from the user and print all the characters of the word.

Q 11. Write a program to find the absolute value of any number.

Q 12. Write a program to sort 3 numbers.

Q 13. Write a program to check whether the number is divisible by other number.

Q 14. Write a program for summation of series.

Q15. Write a program to calculate the factorial of a given number.

ANSWERS

Answer 1:

'else' is used along with 'if' to define the alternative path of condition mentioned in the 'if 'statement is incorrect. This means that only one statement can be evaluated using the if...elif statement. If more than one statement if need to be evaluated than 'elif' used.

Answer 2:

```
x = 1
while x < 100:
  print(x)
  x += 2
```

Answer 3:A loop that never ends unless broken manually.

```
while True:
        print("Looping forever")
```

Answer 4:

```
for i in range(10, 0, -1):
        print(i)
```

Answer 5: range(1, 21)

Answer 6: Break immediately exits the loop, regardless of the loop condition. Continue skips the remaining code in the current iteration and moves to the next iteration.

Answer 7: elif

```
#movie ticket booking
age = int(input("Enter your age: "))
if age < 5:
  print("Ticket is FREE (Child under 5)")
elif age < 13:
  print("Ticket price: $5 (Child)")
elif age < 60:
  print("Ticket price: $10 (Adult)")
else:
  print("Ticket price: $7 (Senior Citizen)")
```

Answer 8: Nothing will be printed because the condition i < 8 is false initially.

Answer 9: Output: 0

Answer 10:

```
s=input("enter word")
for i in s:
  print(i)
```

Answer 11:

```
#program to find the absolute number
num = float(input("Enter a number: "))
if num < 0:
  absolute_value = -num
else:
  absolute_value = num
print("The absolute value is:", absolute_value)
```

Answer 12:

```
#program to sort three number
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))
c = int(input("Enter third number: "))
if a > b:
  a, b = b, a
if b > c:
  b, c = c, b
if a > b:
  a, b = b, a
print("Numbers in ascending order:", a, b, c)
```

Answer 13:

```
#check if a number is divisible by another
num1 = int(input("Enter the first number: "))
div = int(input("Enter the divisor "))
if div== 0:
  print("Divisor cannot be zero.")
elif  num1 % div == 0:
    print(num1 ,"is divisible by ",div)
else:
    print(num1,"is not divisible by",div)
```

Answer 14:

```
#summation of series
n=int(input("enter a number:"))
sum=0
for i in range(1,n+1):
  sum+=i
print("sum of first",n,"number is",sum)
```

Answer 15:

```
#finding factorial of given number
n=int(input("enter a  positive number:"))
factorial=1
for i in range(1,n+1):
  factorial*=i
print("factorial of",n,"is:",factorial)
```

LONG ANSWER QUESTIONS

Q 1.Write a program that reads two numbers and an arithmetic operator and displays the computed result.

Q2 .Write a program to print the sum of natural numbers between 1 to 7.

Q 3.Write a program to print first 10 elements of the Fibonacci Series.

Q 4. Write a program to print the following pattern :

5 4 3 2 1
5 4 3 2
5 4 3
5 4
5

Q5 .Write a program for calculating grade using if elif.

ANSWERS

Answer 1 :

```
num1 = float(input("Enter the first number: "))
operator = input("Enter an operator (+, -, *, /): ")
num2 = float(input("Enter the second number: "))
if operator == '+':
  result = num1 + num2
elif operator == '-':
  result = num1 - num2
elif operator == '*':
  result = num1 * num2
elif operator == '/':
  if num2 != 0:
    result = num1 / num2
  else:
    result = "Error: Division by zero!"
else:
  result = "Invalid operator!"
print("Result:", result)
```

Answer 2:

```
#proram to print the sum of natural numbers from 1 to 7
start=1
end=7
total=0
for number in range(start,end+1):
  total+=number
print("The sum of natural number from 1 to 7 is:",total)
```

Answer 3:

```
#proram to print the first 10 elements of the fibonacci series
a, b = 0, 1
print("First 10 Fibonacci numbers:")
for i in range(10):
   print(a, end=' ')
   a, b = b, a + b
```

Answer 4:

```
# Loop from 5 down to 1
for i in range(5, 0, -1):
  for j in range(5, 5 - i, -1):
    print(j, end=' ')
  print()
```

Answer 5:

```
#Grading System
marks = float(input("Enter your marks (0-100): "))
if marks < 0 or marks > 100:
  print("Invalid marks. Please enter a value between 0 and 100.")
elif marks >= 90:
  print("Grade: A")
elif marks >= 80:
  print("Grade: B")
elif marks >= 70:
  print("Grade: C")
elif marks >= 60:
  print("Grade: D")
else:
  print("Grade: F")
```

STRING

Introduction:

A **string** in Python is a sequence of characters enclosed in either single quotes ' ', double quotes " ", or triple quotes ''' ''' / """ """ (for multi-line strings).

```
name = "Ananya"
message = 'Hello, world!'
paragraph = """This is
a multi-line string."""
```

Immutability:

Python strings are immutable data type, meaning once a string is created, it cannot be changed.

```
>>> word = "hello"
>>> word[0] = "H"
Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    word[0] = "H"
TypeError: 'str' object does not support item assig
nment
```

Indexing:

Indexing in strings refers to accessing individual characters of a string using their position (index).

1. Positive Indexing or Forward Indexing:

Indexing starts at 0 for the first character.

```
text = "hello"
print(text[0])
print(text[1])
```

```
h
e
```

2. Negative Indexing or Backward Indexing:

Indexing can also go from the end, using negative numbers.

```
text = "hello"
print(text[-5])
print(text[-1])
```

```
h
o
```

String Operations in Python:

- **Concatenation**
- **Repetition**
- **Membership Testing**
- **Slicing**

These operations are essential for manipulating and working with strings effectively in programming.

Concatenation: Concatenation is the operation of joining two or more strings together. In Python, this can be achieved using the + operator.

```
>>> string1 = "computer"
>>> string2 = "science"
>>> mystring = string1 + " " + string2
>>> mystring
'computer science'
```

```
>>> num1=12
>>> num2= "abc"
>>> num1+num2
Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    num1+num2
TypeError: unsupported operand type(s) for +: 'int'
and 'str'
```

```
>>> num1="12"
>>> num2= "abc"
>>> num1+num2
'12abc'
```

Repetition: Repetition allows you to create a new string by repeating an existing string a specified number of times. This is done using the * operator.

```
>>> string = "Hello"
>>> num1="10"
>>> result1=string*3
>>> result1
'HelloHelloHello'
```

```
>>> Result2=num1*4
>>> Result2
'10101010'
```

Membership: Membership operator checks whether a substring exists within a string. This can be done using the **in** and **not in** keywords.

```
>>> text = "learning Python is fun"
>>> "Python" in text
True
```

```
>>> text = "learning Python is fun"
>>> "fun" not in text
False
```

Slicing: In Python, slicing enables you to extract a subset of a string by defining the start and end indices, and optionally a step value.
The syntax for slicing is `string[start:stop:step]`
Example: `text= "hello world"`

Sample Code	Description	Output
`substring1 = text[0:5]` `print(substring1)`	extracts character from index 0 to 4	hello
`substring2 = text[6:11]` `print(substring2)`	extracts character from index 6 to 10	world
`substring3 = text[:5]` `print(substring3)`	extracts character from **beginning** (index 0) to index 4	hello
`substring4 = text[7:]` `print(substring4)`	extracts character from index 7 to end of the string	orld
`substring5 = text[::2]` `print(substring5)`	extracts every other character from the string text	hlowrd
`Substring6 = text[-5:-1]` `print(substring6)`	starts at the 5th character from the end, up to but not including last character	worl
`Substring7 = text[::-1]` `print(substring7)`	Gives string in reverse order	dlrow olleh
`Substring8 = text[5:5]` `print(Substring8)`	No error but also no answer	
`Substring9 = text[::0]` `print(Substring9)`	ValueError: slice step cannot be zero	ValueError
`SubString10 = text[1:6][1:3]` `Print(Substring10)`	First slices text at 1 to 5 and then slices the resultant string from 1 to 2	ll

TRAVERSING A STRING USING LOOPS:

For Loop:

```
a="best"
for i in range(len(a)):
    print(a[i])
```

```
a="best"
for i in a:
    print(i)
```

While Loop:

```
a="best"
i=0
while i<len(a):
  print(a[i])
  i+=1
```

Both Loops gives the same output:

```
b
e
s
t
```

Built-in functions in string:

Function	Description	Example

len( )	Returns the length of the given string	>>> str1 = 'Hello World!'
		>>> len(str1)
		12

capitalize( )	Returns the string with first letter of the string in uppercase and rest in lowercase	>>> str1 = 'hello WORLD!'
		>>> str1.capitalize()
		'Hello world!'

title( )	Returns the string with first letter of every word in the string in uppercase and rest in lowercase	>>> str1 = 'hello WORLD!'
		>>> str1.title()
		'Hello World!'

lower( )	Returns the string with all uppercase letters converted to lowercase	>>> str1 = 'hello WORLD!'
		>>> str1.lower()
		'hello world!'

upper( )	Returns the string with all lowercase letters converted to uppercase	>>> str1 = 'hello WORLD!'
		>>> str1.upper()
		'HELLO WORLD!'

count(str, start, end)	Returns number of times substring occurs in the given string. If we do not give start index and end index then searching starts from index 0 and ends at length of the string.	>>> str1 = 'Hello World! Hello Hello'
		>>> str1.count('Hello',12,25)
		2
		>>> str1.count('Hello')
		3

find(str,start, end)	It gives you the **index** (position) of the **first time** the substring appears. If we do not give start index and end index then it starts from the beginning (index 0) and searches till the end of the string.If the substring is not found, it returns -1.	>>> str1 = 'Hello World! Hello Hello'
		>>> str1.find('Hello',10,20)
		13
		>>> str1.find('Hello',15,25)
		19
		>>> str1.find('Hello')
		0
		>>> str1.find('Hee')
		-1

index(str, start, end)	Same as find() but raises an exception if the substring is not present in the given string.	>>> str1 = 'Hello World! Hello Hello'
		>>> str1.index('Hello')
		0
		>>> str1.index('Hee')
		ValueError: substring not

		found

endswith()	returns True if the given string ends with the supplied substring otherwise returns False.	>>> str1 = 'Hello World!'
		>>> str1.endswith('World!')
		True
		>>> str1.endswith('!')
		True
		>>> str1.endswith('lde')
		False

startswith( )	returns True if the given string starts with the supplied substring otherwise returns False.	>>> str1 = 'Hello World!'
		>>> str1.startswith(' Hello')
		True

isalnum( )	Returns True if characters of the given string are either alphabets or numeric. If white-space or special symbols are part of the given string or the string is empty it returns False.	>>> str1 = 'HelloWorld'
		>>> str1.isalnum()
		True
		>>> str1 = 'HelloWorld2'
		>>> str1.isalnum()
		True
		>>> str1 = 'HelloWorld!!'
		>>> str1.isalnum()
		False

isalpha( )	It returns True if the string contains only alphabets, otherwise false.	>>>str1="12345" >>>str1.isalpha() **False** >>>str2 ="abc" >>>str2.isalpha() **True**
isdigit( )	It returns True if the string contains only digits, otherwise false.	>>>str1="12345" >>>str1.isdigit() **True** >>>str2 ="1234a5" >>>str2.isdigit() **False**

islower( )	Returns True if the string is non-empty and has all lowercase letters, or has at least one character as lowercase alphabet and rest are non-alphabet characters	>>> str1 = 'hello world!'
		>>> str1.islower()
		True
		>>> str1 = 'hello 1234'
		>>> str1.islower()
		True
		>>> str1 = 'hello ??'
		>>> str1.islower()
		True
		>>> str1 = '1234'
		>>> str1.islower()
		False
		>>> str1 = 'Hello World!'
		>>> str1.islower()
		False12

isupper( )	Returns True if the string is non-empty and has all uppercase letters, or has at least one character as uppercase alphabet and rest are non-alphabet characters	>>> str1 = 'HELLO WORLD!'
		>>> str1.isupper()
		True
		>>> str1 = 'HELLO 1234'
		>>> str1.isupper()
		True

		>>> str1 = 'HELLO ??'
		>>> str1.isupper()
		True
		>>> str1 = '1234'
		>>> str1.isupper()
		False
		>>> str1 = 'Hello World!'
		>>> str1.isupper()
		False

isspace( )	Returns True if the string is non-empty and all characters are white spaces (blank, tab,
	newline, carriage return)
		>>> str1 = ' \n \t \r'
		>>> str1.isspace()
		True
		>>> str1 = 'Hello \n'
		>>> str1.isspace()
		False

lstrip( )	Returns the string after removing the spaces only on the left of the string	>>> str1 = ' Hello World!'
		>>> str1.lstrip()
		'Hello World!'

rstrip( )	Returns the string after removing the spaces only on the right of the string	>>> str1 = ' Hello World!'
		>>> str1.rstrip()
		' Hello World!'

strip( )	Returns the string after removing the spaces both on the left and the right of the string	>>> str1 = ' Hello World!'
		>>> str1.strip()
		'Hello World!'

replace(oldstr, newstr)	Replaces all occurrences of old string with the new string .	>>> str1 = 'Hello World!'
		>>> str1.replace('o','*')
		'Hell* W*rld!'
		>>> str1 = 'Hello World!'
		>>>
		str1.replace('World','Countr
		y')
		'Hello Country!'
		>>> str1 = 'Hello World! Hello'
		>>>
		str1.replace('Hello','Bye')
		'Bye World! Bye'

join( )	The string join() method returns a string by joining all the elements of an iterable (list, string, tuple), separated by the given separator.	>>> str1 = ('HelloWorld!')
		>>> str2 = '-'
		>>> str2.join(str1)
		'H-e-l-l-o-W-o-r-l-d-!'

partition( )	Partitions of the given string at the first occurrence of the substring returns the string partitioned into three parts.
	1. Substring before the separator
	2. Separator
	3. Substring after the separator
	If the separator is not found in the string, it returns the whole string itself and two empty strings
		>>> 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	>>> str1 = 'India is a Great

	the specified substring. If no delimiter is given then words are separated by space.	Country'
		>>> str1.split()
		['India','is','a','Great', 'Country']
		>>> str1 = 'India is a Great Country'
		>>> str1.split('a')
		['Indi', ' is ', ' Gre', 't Country']

QUESTION BANK

MULTIPLE CHOICE QUESTIONS

Q1. Which of the following is String literal?

(A) “ABC” (B)“123” (C)Both of the above (D)None of the above

Q2. What will be the result of the following command:-

>>>"i love python".capitalize()

(A) 'I Love Python' (B) 'I love python' (C) 'I LOVE PYTHON' (D) None of the above

Q3. What will be the output of the following command:

>>>string= "A quick fox jump over a lazy fox"

>>>string.find("fox",9,34)

(A) 29 (B) 28 (C) 8 (D) 7

Q4. Raman is using string.isspace() in his command, Help him to choose the correct option.

>>>string=” Gagan Pandey”

>>>string.isspace( )

(A) returns True (B) returns False (C) Error (D) None of the above

Q5. Find the length of the following string using len( ) :

>>>str= “World in Shock.”

(A) 14 (B) 15 (C) 16 (D) 17

Q6. What will be the output of the following Code:

>>>"India is my Country. I love my Country India".replace("Country","India")

(A) India is my Country. I love my Country India".

(B) Country is my Country. I love my Country Country

(C)'India is my India. I love my India India'

(D) ERROR

Q7. Write the correct code for the following output:-

Output: 'WORLD'

(A) >>>str="$"
>>> join("WORLD",’$’)

(B) >>>str="$"
>>>str.join("WORLD")

(C) >>> "$" = str
>>> join("WORLD",’$’)

(D) >>>str="$"
>>> join("WORLD").str

Q8. The data type of x in the below code is:

txt = "I could eat bananas all day"

x = txt.partition("bananas")

(A) string (B) List (C) Tuple (D) Dictionary

Q9 .The number of element return from the partition function is :

>>>“I love to study python”.partition("study")

(A) 1 (B) 3 (C) 4 (D) 5

Q10. Which of the following methods can be used to check if a string ends with a particular suffix.

(A) end() (B) endswith() (C) finish() (D) startswith()

Q 11.Which operator is used for string concatenation in Python?

(A) * (B) + (C) & (D) %

Q 12. What will be the result of : "abc123".isalnum()

(A) True (B) False (C) Error (D) None

Q 13. What does the following return?

```
str1 = "hello"
str2 = "HELLO"
str1 == str2
```

(A) True (B) False (C) Error (D) None

Q14. What is the output of the following?

```
s = "hello world"
print(s.title())
```

(A) Hello World (B) Hello world (C) HELLO WORLD (D) hello World

Q15. What is the output of:

```
print("apple,banana,grape".split(","))
```

(A) ['apple banana grape'] (B) ['apple', 'banana', 'grape']
(C) ('apple', 'banana', 'grape') (D) Error

ANSWERS

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
C	B	A	B	B	C	B	C
9	**10**	**11**	**12**	**13**	**14**	**15**	
B	B	B	A	B	A	B	

ASSERTION AND REASONING QUESTIONS

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

A. Both A and R are true, and R is the correct explanation of A.
B. Both A and R are true, but R is not the correct explanation of A.
C. A is true but R is false.
D. A is false but R is true.

Question 1:
Assertion (A): Strings in Python are stored as sequences of Unicode characters.
Reason (R): The ord() function can be used to get the ASCII value of a character.

Question 2 :
Assertion (A): The expression "python"[0] = "j" will raise an error.
Reason (R): Strings in Python are immutable.

Question 3 :
Assertion(A): """A Sample Python String""" is a valid Python String.
Reason(R): Triple Quotation marks are not valid in Python.

Question 4 :
Assertion (A): The len() function can be used on strings.
Reason (R): It returns the number of characters present in the string.

Question 5 :
Assertion (A): The string "abc" is not equal to "ABC" in Python.
Reason (R): Python string comparisons are case-insensitive.

ANSWERS

1	2	3	4	5
B	A	C	A	C

SHORT ANSWER QUESTIONS

Q1. What is a string in Python? How can you create a string in Python? Give examples.
Q2. Differentiate between immutable and mutable objects with reference to strings.
Q3. Write a Python statement to convert a string to uppercase.
Q4. What is the difference between **isalpha()** and **isdigit()**?
Q5. Write a Python statement to count how many times a letter appears in a string.

Q6. What is difference between find( ) and index( ) string methods. Give an example to illustrate difference.
Q7. What is the output of "Hello"[1:4]**?**
Q8. Suggest appropriate functions for the following tasks –
(a) To check whether the string contains digits.
(b) To find the occurrence of a string within another string.
(c) To convert the first letter of a string to uppercase.
(d) To convert all the letters of a string to uppercase.
(e) To check whether all the letters of the string are in capital letters.
(f) To remove all the white spaces from the beginning of a string.
Q9. Find and write the output of the following python code:

```
x = "abcdef"
i = "a"
while i in x:
        print(i, end = " ")
```

Q10 . Write the output of following code:

```
first= “Programming Language Learning work”
print(first.partition(“Language”))
```

ANSWERS

Ans 1: A string is a sequence of characters enclosed within single, double, or triple quotes.

```
Example:
s1 = 'Hello'
s2 = "World"
s3 = '''Triple quotes are also valid'''
```

Ans 2: Strings are immutable, meaning they cannot be changed after creation.

```
Example:   s = "Hello"
s[0] = "h" # This will cause an error
```

Ans 3:

```
s = "hello"
print(s.upper())  # Output: HELLO
```

Ans 4: isalpha() is for alphabet-only strings, and isdigit() is for digit-only string

Ans 5:

```
s = "banana"
print(s.count("a"))  # Output: 3
```

Ans 6 : The find() method returns the index of the first occurrence of the substring. If the substring is **not found**, it returns **-1**.
The index() method also returns the index of the first occurrence of the substring. However, if the substring is **not found**, it raises a **ValueError**.

Example:

```
text = "hello world"
print(text.find("world"))   # Output: 6
print(text.find("Python"))  # Output: -1

print(text.index("hello"))  # Output: 0
print(text.index("Python")) # Raises ValueError: substring not found
```

Ans 7 : Output: 'ell' (substring from index 1 to 3)
Ans 8: (a) isalnum() (b) find() (c) capitalize() (d) upper() (e) isupper() (f) lstrip()
Ans 9 : Ans: aaaaaa ---- OR infinite loop
Ans 10: ('Programming ', 'Language', ' Learning work')

LONG ANSWER QUESTIONS

Q 1. Write a program to input a string and calculate the length of each word present in string and print length along with the word.

Q 2. Write a program to input a string and print the total number of uppercase and lowercase letters in a given string.

Q 3. Write a program to accept a word from the user and display in the following pattern.
If the word is "river" then it should display as shown below :

```
r
ri
riv
rive
river
```

Q 4. Find the output of the following code :

```
s = "welcome2cs"
n = len(s)
m = ""
for i in range(0, n):
      if (s[i] >= 'a' and s[i] <= 'm'):
              m = m + s[i].upper()
      elif (s[i] >= 'n' and s[i] <= 'z'):
              m = m +s[i-1]
      elif (s[i].isupper()):
              m = m + s[i].lower()
      else:
              m = m +'&'
print(m)
```

Q 5. Write a program to accept a string that counts how many **digits**, **alphabets**, **spaces**, and **special characters** are present.

ANSWERS

Answer 1:

```
sentence = input("Enter a sentence: ")
words = sentence.split()
print("\nLength of each word:")
for i in words:
    print("words",i,"length",len(i))
```

Answer 2:

```
text = input("Enter a string: ")
u_count = 0
l_count = 0
for c in text:
    if c.isupper():
        u_count += 1
    elif c.islower():
        l_count += 1
print("\nTotal uppercase letters:", u_count)
print("Total lowercase letters:", l_count)
```

Answer 3:

```
str1=input("enter any string")
str2=""
for i in str1:
     str2=str2+i
     print(str2)
```

Answer 4:

sELCcME&Cc

Answer 5:

```
text = input("Enter a string: ")
alphabet_count = 0
digit_count = 0
space_count = 0
special_count = 0
for i in range(len(text)):
    c = text[i]
    if c.isalpha():
        alphabet_count += 1
    elif c.isdigit():
        digit_count += 1
    elif c.isspace():
        space_count += 1
    else:
        special_count += 1
print("Total alphabets:", alphabet_count)
print("Total digits:", digit_count)
print("Total spaces:", space_count)
print("Total special characters:", special_count)
```

LIST

- Written in **Square [ ] brackets,** a list is a collection of comma-separated values.
- Lists are used to **store multiple items** in a single variable.
- When we say that lists are **ordered**, it means that the items have a defined order, and that order will not change.
- The list is **mutable**, meaning Lists can be modified after creation, we can change, add, and remove items in a list after it has been created.
- Elements in a list are **indexed** with integers starting from 0. Since lists are indexed, lists can have duplicate values, items with the same value.
- **Heterogeneous** - Lists can contain elements of different data types. For example, a list can contain integers, strings, floats, and even other lists.
- **Built-in Methods** - Python lists come with built-in methods for various operations like sorting, reversing, searching, etc., making them versatile for a wide range of tasks.
- **Iterable** - Lists can be used in iterations using loops (e.g., for loop)
- **Slicing** - Lists support slicing operations, allowing you to extract sublists by specifying a range of indices.

List Creation and Initialization: Giving Initial value to list

```
#List Initialization

# Empty List Way 1
list1 = list()
print(type(list1))

# Empty List Way 2
list1 = []
print(type(list1))

# Creation of list with  homogenous elements
list1 = ["A","B","C","D"]
print("list1:",list1)
list2 = [10,20,30,40,50]
print("list2:",list2)
list3 = ["Delhi","Mumbai","Chennai","Kolkata"]
print("list3:",list3)

# Creation of list with  heterogenous elements
list4=["Aman",18,95.5,11,"A"]
print("list4:",list4)

# Creation of Nested List
list5=[10,20,30,[40,50],60,70]
print("list5:",list5)

# Creation of list taking input from user
list6=eval(input("Enter List element in box bracket:"))
print("list6:",list6)
```

OUTPUT:

```
<class 'list'>
<class 'list'>
list1: ['A', 'B', 'C', 'D']
list2: [10, 20, 30, 40, 50]
list3: ['Delhi', 'Mumbai', 'Chennai', 'Kolkata']
list4: ['Aman', 18, 95.5, 11, 'A']
list5: [10, 20, 30, [40, 50], 60, 70]
Enter List element in box bracket:[10,20,30,40,50]
list6: [10, 20, 30, 40, 50]
```

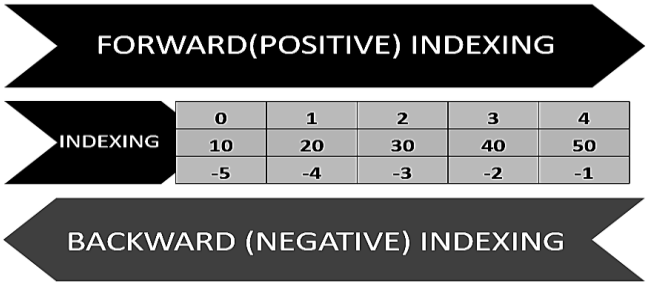
INDEXING: Each character of a List can be accessed by two types of indexing.

- **Forward indexing:** First element of a List has an index 0 and next has 1 and so on.
- **Backward indexing:** Last element of the List is having an index of -1 and second last has -2 and so on.

```
my_list = [10, 20, 30, 40, 50]
print(my_list[0])
print(my_list[-1])
```

OUTPUT:

```
10
50
```



LIST ARE MUTABLE: Value of a list can be changed by either

1. **Using indexing**: updating single index value
2. **Using Slicing**: updating multiple values

OUTPUT:

```
# Using Indexing
list1=[10, 20, 30, 40, 50, 60]
list1[3]=80
print(list1)
```

```
[10, 20, 30, 80, 50, 60]
```

OUTPUT:

```
# Using Slicing
list1=[10, 20, 30, 40, 50, 60]
list1[2:5]=[80,90,100]
print(list1)
```

```
[10, 20, 80, 90, 100, 60]
```

LIST OPERATION: Concatenation, Repetition, Membership and Slicing

```
# Concatenation
list1=[10, 20, 30]
list2=[40, 50, 60]
print(list1+list2)
```

```
[10, 20, 30, 40, 50, 60]
```

```
# Repetition
lst=["Pop", 20]
print(lst*2)
```

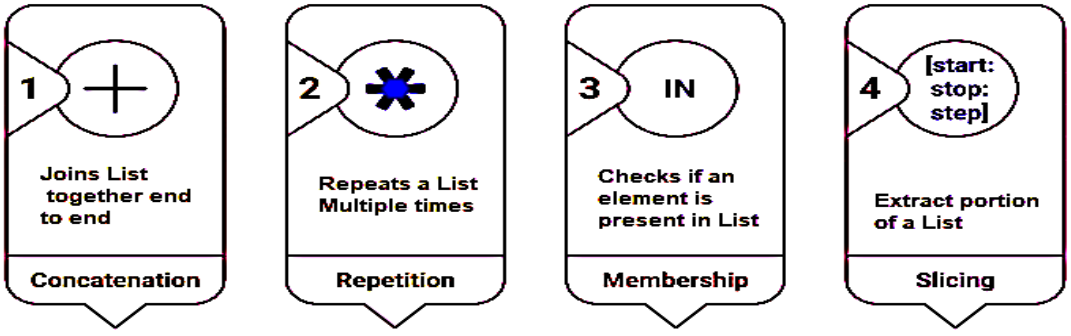
```
['Pop', 20, 'Pop', 20]
```

```
# Membership
list1=[10, 20, 30]
print(10 in list1)
```

```
True
```

```
# Slicing
l1=[10, 20, 30, 40, 50, 60]
print(l1[1:5:2])
```

```
[20, 40]
```



SLICING IN LIST

Note: *Slicing never gives IndexError*
Syntax: list_object_name[start : stop : interval]

- Here start, stop and interval all have default values.
- Default value of start is 0, stop is n+1 (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:

```
lst=[4,7,5,2,10]
lst[1:5:2]
```

OUTPUT:

```
[7, 2]
```

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

```
lst=[4,7,5,2,10]
print(lst[5:0:-2])
print(lst[-1:-4:-1])
```

OUTPUT:

```
[10, 5]
[10, 2, 5]
```

LIST TRAVERSAL

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 by using while loop and for loop

- Using **while loop:** A while loop can also be used for list traversal, requiring manual index management.

```
my_list = [10, 20, 30, 40, 50]
i = 0
while i < len(my_list):
    print(my_list[i])
    i += 1
```

OUTPUT:

```
10
20
30
40
50
```

- Using a **for loop**: A for loop iterates through each element of a list directly, executing a block of code for every element.

```
my_list = [10, 20, 30, 40, 50]
for element in my_list:
    print(element)
```

OUTPUT:

```
10
20
30
40
50
```

- Using **for loop** with **range()**: Combining a for loop with the range() function allows iterating through the list using indices.

```
lst=[4,7,5,2,10]
for i in range(len(lst)):
    print(lst[i])
```

OUTPUT:

```
4
7
5
2
10
```

Built In Function: len(), list(), append(), extend(), insert(), count(), index(), remove(), pop(), reverse(), sort(), sorted(), min(), max(), sum();

List Insertion: Adding element in list

1. **append ():** Adding element at the last index number of list
2. **extend ():** Adding multiple elements at the end of list
3. **insert ():** Adding element at the specified position

```
# List Insertion

# Adding new element in the last  - Single unit
list1 = [10,20,30,40,50]
list1.append(60)
print(list1)

list1.append([70,80])     #creates nested list
print(list1)

# Adding new elements in the last  - multiple unit
list1.extend([90,100])
print(list1)

# Adding a new element on a specific Index

list1.insert(4,45)
print(list1)
```

OUTPUT:

```
[10, 20, 30, 40, 50, 60]
[10, 20, 30, 40, 50, 60, [70, 80]]
[10, 20, 30, 40, 50, 60, [70, 80], 90, 100]
[10, 20, 30, 40, 45, 50, 60, [70, 80], 90, 100]
```

List Deletion: removing element from list

1. **remove ():** deleting element by value
2. **pop ():** Removes and returns last element from the end or from specified index
3. **del:** delete one or more than one element either using indexing or using slicing

- remove any element by value: **remove ()**

```
# remove function
list1=[10, 20, 30, 40, 50, 60]
list1.remove(30)
print(list1)

[10, 20, 40, 50, 60]
```

OUTPUT:

- removing last element: **pop ()**

```
# removing last element using pop function
list1=[10, 20, 30, 40, 50, 60]
a=list1.pop()
print(list1)
print("Item deleted",a)
```

OUTPUT:

```
[10, 20, 30, 40, 50]
Item deleted 60
```

- removing element from a specific index: **pop(index)**

```
# removing element by index using pop function
list1=[10, 20, 30, 40, 50, 60]
a=list1.pop(4)
print(list1)
print("Item deleted",a)
```

OUTPUT:

```
[10, 20, 30, 40, 60]
Item deleted 50
```

- removing one or multiple values by index: **del**

```
# using del command
list1=[10, 20, 30, 40, 50, 60]
del list1[2]
# remove element at index2(30)
print(list1)

del list1[1:4]
# removes all elements from index1 to 4(20,40,50)
print(list1)
```

OUTPUT:

```
[10, 20, 40, 50, 60]
[10, 60]
```

OTHER FUNCTION

Global functions: Since List is a sequence the following built-in functions can also be used on list.

Examples: len( ), sorted( ), min( ), max( ), sum( )

Function	Description	Syntax	Example	Output
len( )	Returns the number of items in a list.	len(list)	len([10, 20, 30])	3
sorted( )	Returns a new sorted list from the elements of the given list.	sorted(list, reverse=False)	sorted([3, 1, 4])	[1, 3, 4]
min( )	Returns the smallest item in the list.	min(list)	min([5, 2, 9])	2
max( )	Returns the largest item in the list.	max(list)	max([5, 2, 9])	9
sum( )	Returns the sum of all items in the list.	sum(list)	sum([1, 2, 3])	6

DATA TYPE SPECIFIC FUNCTIONS: also known as methods, are associated with particular data types. They are called using the dot notation on an object of that data type (e.g.,**object.function()**). These functions perform operations specific to the data type they belong to.

Some list specific functions : count( ), index( ),reverse( ), sort( ),

Function	Description	Syntax	Example	Output
count( )	Returns the number of occurrences of a specified element in the list.	list.count(element)	a=[1, 2, 2, 3] a.count(2)	2
index( )	Returns the index of the first occurrence of a specified element.	list.index(element)	a=['a', 'b', 'c'] a.index('b')	1
reverse( )	Reverses the elements of the list in place.	list.reverse()	a = [1, 2, 3] a.reverse( ) print(a)	[3, 2, 1]
sort( )	Sorts the list in ascending order by default; can be customized.	list.sort(reverse=False)	a = [3, 1, 2] a.sort() print(a)	[1, 2, 3]

SUGGESTED PROGRAMS

Program-1 : Write a program in Python to take a list from user and 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))
mean=sum(L)/len(L)
print("Mean of the numeric values of the list = ",mean)
```

OUTPUT:

```
Enter an integer number to append in the list: 10
Do you want to add more elements in the list (Y/N): Y
Enter an integer number to append in the list: 20
Do you want to add more elements in the list (Y/N): Y
Enter an integer number to append in the list: 30
Do you want to add more elements in the list (Y/N): N
List is:
 [10, 20, 30]
Minimum(Smallest) number of the list =  10
Maximum(Largest) number of the list =  30
Mean of the numeric values of the list =  20.0
```

Program-2 : 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: 40
Element found at index:  3
Do you want to search more item (Y/N): y
Enter the number to be search in the list: 80
Number is not found in the list
Do you want to search more item (Y/N): n
```

Pogram-3 : Write a program in Python to count the frequency of all elements of a list.

```
L= [10, 20, 30, 20, 40, 30, 20, 50]
print("List is: ",L)
for i in L:
    Freq=L.count(i)
    print(i,":",Freq)
```

OUTPUT:

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

QUESTION BANK

	MULTIPLE CHOICE QUESTIONS
1	Which of the following list methods is used to delete element from the list? a)del( ) b) delete( ) c)pop( ) d)All of these
2	Suppose list1 is [1, 3, 2], What is list1 * 2? a) [2, 6, 4]. b) [1, 3, 2, 1, 3]. c) [1, 3, 2, 1, 3, 2] . d) [1, 3, 2, 3, 2, 1]
3	Given the list L=[-1,4,6,-5] ,write the output of print(L[-1:-3]) a)[4,6,-5] b)[ ] c)[6,-5] d)error
4	In list slicing, the start and stop can be given beyond limits. If it is then

	a. raise exception Index Error b. raise exception Value Error c. return elements falling between specified start and stop values d. return the entire list
5	Which of the following statement is true for extend( ) method? a. adds element at last b. adds multiple elements at last c. adds element at specified index d. adds elements at random index
6	The statement del l[1:3] does which of the following task? a. deletes elements 2 to 4 elements from the list b. deletes 2nd and 3rd element from the list c. deletes 1st and 3rd element from the list d. deletes 1st, 2nd and 3rd element from the list
7	The data type list is an ordered sequence which is _____and made up of one or more elements. a. Mutable b. Immutable c. Both a) and b) d. None of the above
8	We can access each element of the list or traverse a list using a. a. for loop b. while loop. c. Both a) and b) d. None of the above
9	_____function returns the element whose index is passed as parameter to this function and also removes it from the list. a. push( ) b. remove( ) c. pop( ) d. None of the above
10.	Suppose list1 is [4, 2, 2, 4, 5, 2, 1, 0], which of the following is correct syntax for slicing operation? a) print(list1[2:]) b) print(list1[:2]) c) print(list1[:-2]) d) all of the above

ANSWERS

1	2	3	4	5	6	7	8	9	10
c	c	b	c	b	b	a	c	c	d

ASSERTION AND REASONING QUESTIONS

	In the following questions, a statement of Assertion(A) is followed by a statement of Reason (R). Make 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 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
1	Assertion : len(L) statement returns the total number of elements in the list. Reason : count( ) method is used to find the total number of elements in the list.
2	Assertion: Any comma-separated group of values creates a list. Reason: Only a group of comma-separated values or expressions enclosed in [ ] , creates a list
3	Assertion: Lists and strings have similar types of indexing. Reason: Both lists and strings have two-way indexing , forward indexing and backward indexing.
4	Assertion :Lists are similar to strings in a number of ways like indexing , slicing and accessing individual elements. Reason : Lists, unlike strings , are mutable
5	Assertion: A list slice is an extracted part of a list. Reason: A list slice is a list in itself.

6	Assertion: List has indices to access the elements. Reason: The list is an ordered data type.
7	Assertion: List traversal in Python is done using for and while loop. Reason: Traversal can be done only through forward indexing.
8	Assertion: In Python, list is an immutable collection of data. Reason: Mutable means that any change or alteration in data is mentioned in the same place. The updated collection will use the same address for its storage.
9	Assertion: The position of each element in the list is considered as its index. Reason: Indexing in a list can be positive and negative; index is defined using [ ] brackets.
10	Assertion: sort( ) method sorts the objects of list in ascending order. Reason: Defining sort( ) method with reverse=True as an argument sorts the list elements in descending order.
11	Assertion: List slices are the sub-part of a list extracted out. Reason: In Python, Slice operator[:] is used to select a range of elements from a sequence.
12	Assertion: Indexing refers to accessing elements of a sequence. Python offers two types of indexing, viz. positive or forward and negative or backward indexing. Reason: Both forward and backward indexing are implemented in all the sequences, which start with first (1st) index.

ANSWERS

1	2	3	4	5	6	7	8	9	10	11	12
c	d	a	a	a	a	c	d	a	a	a	c

SHORT ANSWER QUESTIONS

1	What will be the output of the following python program: data = [10,20,30, 60,70] data[3:3]=[40,50] print(data) data.pop(3) print(data) data.extend([10,20]) print(len(data))
2	What will be the output of the following list operations? data = [10,20,30,[40,50,60],[70,80]] print(data[3]+data[-1]) print(data[-2][-2])
3	Ganga is learning to use python Lists. Help her to get the answers of the following operations based on the given list: data = [10,20,30] data[1:3]=[5,10] print(data) data.extend([3,4]) x =data.count(10) print(data[x:]) data.sort() print(data) print(data.pop(2))
4	Give the output of the following L = [1,2,3,4,5,6,7,8,9] print(L[: -1]) print(L[: :-1])

5	Find the output of the following S = 'abcdefgh' L = list(S) print(L[1:4])
6	Give the output of the following L = [1,2,3,4,5,6,7,8,9] print(L.count(2)) print(L.index(2)
7	Predict the output with respect to the list L=[40,20,30,10,50] (a) print(L) (b) print(len(L)) (c) L.pop() ; print(L) (d) L.append(70); print(L) (e) L.sort(); print(L)
8	Predict the output: (a) b=[[9,6],[4,5],[7,7]] X=b[:2] X.append(10) print(X) (b) b=[[9,6],[4,5],[7,7]] X=b[:2] X[1].append(10) print(X)
9	What is the difference between pop() and pop(0)?
10	What is the difference between extend() and append()?
	ANSWERS
1	[10, 20, 30, 40, 50, 60, 70] [10, 20, 30, 50, 60, 70] 8
2	[40, 50, 60, 70, 80] 50
3	[10, 5, 10] [10, 3, 4] [3, 4, 5, 10, 10] 5
4	[1, 2, 3, 4, 5, 6, 7, 8] [9, 8, 7, 6, 5, 4, 3, 2, 1]
5	['b', 'c', 'd']
6	1 1
7	a) [40, 20, 30, 10, 50] b) 5 c) [40, 20, 30, 10] d) [40, 20, 30, 10, 70] e) [10, 20, 30, 40, 70]
8	(a) [[9, 6], [4, 5], 10] (b) [[9, 6], [4, 5, 10]]
9	pop(): Removes and returns the last element of the list. Equivalent to pop(-1). pop(0): Removes and returns the first element of the list (index 0).
10	append(): Adds a single element to the end of the list. If we append a list, it will be added as a nested list

	extend(): Adds elements of an iterable (like a list, tuple, set) to the end of the list. It unpacks the iterable and adds each item individually.

LONG ANSWER QUESTIONS

1	Write the most appropriate list method to perform the following task in a given list **my_list**: (a) Add element at 4th position in the given list (b) Add element in the last position of the list (c) Delete 2nd element from the given list (d) Delete the given element from the given list (e) Add elements (more than one at the end of the given list)
2	Start with the list [8,9,10]. Do the following using list functions: (a) Set the second entry (index 1) to17 (b) Add 4,5 and 6 to the end of the list. (c) Remove the first entry from the list. (d) Sort the list (e) Print the list element twice. (f) Insert 25 at index 3
3	Write a program to create a list of elements L and then create another list named 'indexList' that stores the indices of all Non-Zero Elements of L. For example: If L contains [12,4,0,11,0,56] Then indexList will have - [0,1,3,5]
4	Write a program in python, which accepts a list of numbers and a numeric value by which all elements of the list are shifted to left. Sample Input data of the list Lst=[10,20,30,40,50,12,11] and n=2 Output Lst:[30,40,50,12,11,10,20]
5	Write a program in Python, which accepts a list ***num*** of integers, and interchange the adjacent elements of the list and print the modified list as shown below: (Number of elements in the list is assumed as even) Original List: num = [5,7,9,11,13,15] After Rearrangement num = [7,5,11,9,15,13]
6	Write a python program to find the largest element in a list and then reverse the list contents and display it. Don't use in-built functions for the program.
7	Write a Python program that accepts a list L of numbers and finds the sum of all even numbers and the sum of all odd numbers.. If L=[1,2,3,4] Output : Sum of even nos:6 Sum of odd nos:4
8	Based on the given sample list write python commands to obtain as asked sample_list = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100] i) Get the first 3 elements in reverse order. ii) Exclude the first and last elements, then return every second element from the result. iii) Reverse the list and return every third element. iv) Get the 4th, 3rd, and 2nd last elements in the same order. v) Get [90, 70, 50, 30] using slicing only.
9	Write output of the following python commands. sample_list = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

	i) sample_list[:-5:-1] ii) sample_list[2:-1:2] iii) sample_list[1::2] iv) sample_list[9:0:-2] v) sample_list[-2:0:-1][::3]
	ANSWERS
1	(a) insert( ) (b) append( ) (c) pop(2) (d) remove( ) (e) extend( )
2	my_list = [8, 9, 10] (a) Set the second entry (index 1) to 17 my_list[1] = 17 # my_list is now [8, 17, 10] (b) Add 4, 5, and 6 to the end of the list my_list.extend([4, 5, 6]) # my_list is now [8, 17, 10, 4, 5, 6] (c) Remove the first entry from the list my_list.pop(0) # my_list is now [17, 10, 4, 5, 6] (d) Sort the list my_list.sort() # my_list is now [4, 5, 6, 10, 17] (e) Double the list my_list = my_list * 2 # my_list is now [4, 5, 6, 10, 17, 4, 5, 6, 10, 17] (f) Insert 25 at index 3 my_list.insert(3, 25) # my_list is now [4, 5, 6, 25, 10, 17, 4, 5, 6, 10, 17] # Final list #print(my_list) #[4, 5, 6, 25, 10, 17, 4, 5, 6, 10, 17]

3	`L=eval(input("enter list elements"))`
	`indexList=[]`
	`for i in range(len(L)):`
	`  if L[i]!=0:`
	`  indexList.append(i)`
	`print(L)`
	`print(indexList)`

4	`lst=eval(input("Enter list:"))`
	`n=int(input("Enter number by which elements to be shifted:"))`
	`for i in range(n):`
	`  first=lst.pop(0)`
	`  lst.append(first)`
	`print(lst)`

5	num=eval(input("enter list elements"))
	n=len(num)
	if n%2==0:
	  for i in range(0,n,2):
	  num[i], num[i+1] = num[i+1], num[i]
	print(num)

6	L=eval(input("enter list elements"))
	n = len(L)
	max = L[0]
	for i in range(1, n) :
	  if L[i] > max :
	  max =L[i]
	print("Largest element",max)
	
	for i in range(n//2):
	  L[i],L[len(L)-1-i] = L[len(L)-1-i], L[i]#swapping elements
	print('Reverse list:', L)

7	L=eval(input("enter list elements"))
	SE=0
	SO=0
	for i in L:
	  if i%2==0:
	  SE+=i
	  else:
	  SO+=i
	print("Sum of Even nos=",SE)
	print("Sum of Odd nos=",SO)

8	i) sample_list[2::-1] ii) sample_list[1:-1:2] iii) sample_list[::-1][::3] iv) sample_list[-4:-1] v) sample_list[8:1:-2]
9	i) [100, 90, 80, 70] ii) [30, 50, 70, 90] iii) [20, 40, 60, 80, 100] iv) [100, 80, 60, 40, 20] v) [90, 60, 30]

PRACTICE PROGRAMS

1	Write a python program that accepts a list of integers from the user and creates a new list from the existing list containing all the numbers that have three or more digits. Eg: for existing list [10,100, 99,200,1000] the new list should be [100,200,1000]
2	Write a Python program that accepts a list of countries from the user and prints all the countries whose names have more than 5 letters.
3	Write a Python program that accepts a list of integers from the user and prints all the integers that end with 8. Eg: for the list [10, 28, 8, 86, 98] the program should print 28 8 98
4	Create a list k, and put all the odd numbers between m and n into k. Take m and n as input from the user.
5	Write a program to enter a list and a number. Subtract all elements of the list from the entered number, then remove all elements that are less than or equal

	to 0 from the list. Print changed list, number of elements in the list, number of elements removed from the list and sum of elements of changed list.
6	Write a program to get a list of numbers from the user and triple the value of all elements at even indices. For example, if the list entered is [1,2,3,4,5,6] the resultant list should be [3,2,9,4,15,6]
7	Write a program to take the names of teachers as input in a list called Teachers, and then search for a given teacher's name. Display all positions where the name occurs. For example : If the Teachers contain ["Ankit" , "Siddharth", "Rahul " ,"Sangeeta", "rahul"] and TName contains "Rahul" The function should display Rahul at 2 rahul at 4
8	Write a Python program that accepts a list L of numbers and a number to be searched. If the number exists, replace all its occurrences with 0. Otherwise, display an appropriate message. Example : L=[10,20,30,10,40] Number to be searched = 10 List after replacement : [0,20,30,0,40]
9	Write a program to accept a list of numbers and replace all numbers greater than or equal to 33 with "PASS" and all numbers less than 33 with "FAIL". For example, if the input is [22, 55, 65, 27, 92, 33], the output should be ["FAIL", "PASS", "PASS", "FAIL", "PASS", "PASS"].

TUPLES

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

Example:

```
T = ( )                    # Empty Tuple
T = (1, 2, 3)              # Tuple of integers
T = (1, 3.4, 7)            # Tuple of numbers
T = ('a', 'b', 'c')        # Tuple of characters
T = ('A',4.5,'Ram',45)     # Tuple of mixed values
T = ('Amit', 'Ram', 'Shyam') # Tuple of strings
```

Creating a tuple in Python:

1. Create a tuple with three integer elements

```
T= (8, 6, 7)
print(T)
print(type(T))
```

OUTPUT:

```
(8, 6, 7)
<class 'tuple'>
```

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

```
T= (8,)
print(T)
print(type(T))
```

OUTPUT:

```
(8,)
<class 'tuple'>
```

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

```
T= 7,
print(T)
print(type(T))
```

OUTPUT:

```
(7,)
<class 'tuple'>
```

4. Create a tuple without parentheses.

```
T= 7,8,9,10
print(T)
print(type(T))
```

OUTPUT:

```
(7, 8, 9, 10)
<class 'tuple'>
```

NOTE:

- **Value without a comma is a single element is NOT TUPLE**

```
T= (10)
print(T)
print(type(T))
```

OUTPUT:

```
10
<class 'int'>
```

- **Value in Quotes is string, NOT TUPLE**

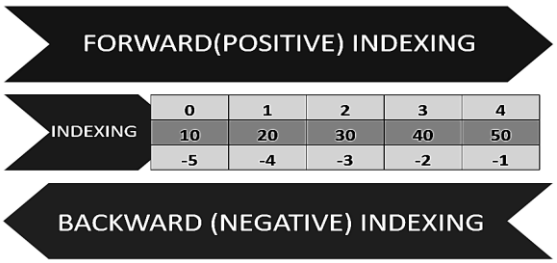
```
T= ("Hello")
print(T)
print(type(T))
```

OUTPUT:

```
Hello
<class 'str'>
```

TUPLE INDEXING

T=(10,20,30,40,50)



Tuple Operations:

1. **Concatenation:** Using + operator

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

OUTPUT:

```
(1, 2, 3, 4, 5, 6)
```

2. **Repetition:** Using * repetition operator, makes multiple copies of a tuple

```
T1=(1,2,3)
T=T1*3
print(T)
```

OUTPUT:

```
(1, 2, 3, 1, 2, 3, 1, 2, 3)
```

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

a. **in**: It return True if an element with the specified value is present in the tuple
b. **not in**: It returns True if an element with the specified value is not present in the tuple.

```
T = (1,"KVS", 8, 6, 5)
print(5 in T)
print("KVS" in T)
print(9 in T)
print(9 not in T)
```

OUTPUT:

```
True
True
False
True
```

TUPLE SLICING

Gives a subset of elements, creating a new tuple without modifying the original. It uses the syntax **tuple_name[start:stop:step].**

start: Index where the slice begins (inclusive). Defaults to 0 if omitted.
stop: Index where the slice ends (exclusive). If omitted, it slices to the end of the tuple.
step: Interval between elements in the slice. Defaults to 1 if omitted.

Example:

my_tuple = (10, 20, 30, 40, 50, 60)
 # Slicing with start and stop
slice1 = my_tuple[1:4] Output: (20, 30, 40)
 # Slicing with start and default stop
slice2 = my_tuple[2:] Output: (30, 40, 50, 60)
 # Slicing with default start and stop
slice3 = my_tuple[:] Output: (10, 20, 30, 40, 50, 60) - creates a copy
 # Slicing with step
slice4 = my_tuple[::2] Output: (10, 30, 50) - every second element
 # Negative indexing and slicing
slice5 = my_tuple[-3:-1] Output: (40, 50)

 # Negative step
slice6 = my_tuple[::-1] Output: (60, 50, 40, 30, 20, 10) - reverses the tuple

DIFFERENCE BETWEEN LIST AND TUPLE

List	Tuple
Elements are enclosed in square brackets i.e. [ ]	Elements are enclosed in parenthesis i.e. ( )
It is mutable data type	It is immutable data type
Iterating through a list is slower as compared to Tuple	Iterating through a tuple is faster as compared to List

Built-in functions/methods:

1. **len( )** function: returns number of elements in the tuple, i.e.,
```
>>> T = (10,20,30,40,50,60,70,80,90)
>>> len(T)
9
```
2. **tuple( )** function: This function creates an empty tuple or creates a tuple if a sequence is passed as argument.
```
>>> L = [10, 20, 30] # List
>>> T = tuple(L) # Creates a tuple from the list
>>> print(T)
(10, 20, 30)
>>> str = "Computer" # String
>>> T = tuple(str) # Creates a tuple from the string
>>> print(T)
('C', 'o', 'm', 'p', 'u', 't', 'e', 'r')
```
3. **count( )** function: returns the count of a member / element (Number of occurrences) in a given tuple.,i.e.,
```
>>> T = (10,20,30,20,20,0,70,30,20)
>>> T.count(20)
4
>>> T = (10,20,30,20,20,0,70,30,20)
>>> T.count(30)
2
```
4. **index( )** function: returns the index of the first occurrence of the element, i.e.,
```
>>> T = (10,20,30,40,50,600,70,80,90)
>>> T.index(50)
4
```
But if the given item does not exist in tuple, it raises ValueError exception.
```
>>> T = (10,20,30,40,50,600,70,80,90)
>>> T.index(55)
 ValueError: tuple.index(x): x not in tuple
```
5. **sorted( )** function: return a new list of sorted elements of the tuple in ascending order (by default). It does not change the existing tuple.
```
>>> T = (10,40,30,78,65,98,23)
>>> X = sorted(T) # Make a list of values arranged in ascending order
>>> print(X)
[10, 23, 30, 40, 65, 78, 98]
```

```
>>> Y = sorted(T, reverse = False) # Make a list of values arranged in ascending order
>>> print(Y)
[10, 23, 30, 40, 65, 78, 98]
>>> Z = sorted(T, reverse = True) # Make a list of values arranged in descending order
>>> print(Z)
[98, 78, 65, 40, 30, 23, 10]
```

6. **min( )** function: returns element having minimum value, i.e.,

```
>>> T = (10,20,30,40,50,600,70,80,90)
>>>min(T)
10
>>> T = ('pankaj','pinki','parul')
>>> min(T)
'pankaj'
```

7. **max( )** function: returns element having maximum value, i.e.,

```
>>> T = (10,20,30,40,50,600,70,80,90)
>>>max(T)
600
>>> T = ('pankaj','pinki','parul')
>>> max(T)
'pinki'
```

8. **sum( )** function: returns the sum of elements of the tuple, i.e.,

```
>>> T = (10,20,30,40,50,600,70,80,90)
>>> sum(T)
990
```

TUPLE ASSIGNMENT (UNPACKING TUPLE)

It allows a tuple of variables on the left side of the assignment operator to be assigned respective values from a tuple on the right side. The number of variables on the left should be same as the number of elements in the tuple.

Example:

```
T = ('A', 100, 20.5)
x,y,z = T
print(x,y,z)
```

OUTPUT:

```
A 100 20.5
```

NESTED TUPLE

A tuple containing another tuple in it as a member is called a nested tuple, e.g.,

```
students = (101,'Punit', (82,67,75,89,90)) # nested tuplelen(students)

print(students[1]) # 2nd element of tuple

print(students[2]) # 3rd element of tuple


print(students[2][3]) # Accessing 4th element of inner tuple
```

Output

```
Punit
(82, 67, 75, 89, 90)
89
```

QUESTION BANK

MULTIPLE CHOICE QUESTIONS

1	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))
2	Suppose tuple1 = (2, 33, 222, 14, 25), What is tuple1[::-1]? a) [25, 14, 222, 33, 2] b) Error c) 25 d) (25, 14, 222, 33, 2)
3	Which operator is used for replication? a) + b) % c) * d) //
4	Rajat wants to delete all the elements from the tuple but keep the tuple variable, which statement he should use to perform this operation- a) t.clear( ) b) del t c) t.remove( ) d)None of these
5	Which of the following operations is not valid for tuples in Python? a) Concatenation b) Indexing c) Appending d) Slicing
6	Shreya wants to add a new tuple t2 to the tuple t1, which statement she should use a) sum (t1,t2) b) t2.add(t1) c) t1+t2 d) None of these
7	_____ function returns the number of elements in the tuple- a)len( ) b) max( ) c) min( ) d)count( )
8	Which of the following statements about tuples in Python is true? a) Tuples are mutable b) Tuples can contain elements of different data types c) Tuples support item assignment d) Tuples are denoted by square brackets [ ]
9	What type will be printed when the following code executes? aTuple = ("Orange") print (type(aTuple)) a) list b) tuple c) array d) string
10	What does the index( ) method do in Python tuples? a) Returns the index of the first occurrence of a specified element b) Returns the index of the last occurrence of a specified element c) Adds an element at the specified index d) Replaces an element at the specified index

ANSWERS

1	2	3	4	5	6	7	8	9	10
b	d	c	d	c	c	d	c	d	a

ASSERTION AND REASONING QUESTIONS

	In the following questions, a statement of Assertion(A) is followed by a statement of Reason (R). Make the correct choice as : a. Both A and R are true and R is the correct explanation of A. b. Both A and R are true but R is not the correct explanation of A. c. A is true but R is false. d. A is false but R is true.
1	Assertion (A): Tuple in Python is an ordered and immutable data type. Reasoning (R): Tuples can contain heterogenous data and permit duplicate values as well.
2	Consider the given two statements: Statement 1: t1 = tuple('python') Statement 2: t1[4] = 'z' Assertion (A): The above code will generate an error.

	Reasoning (R): Tuple is immutable by nature.
3	Assertion(A): Lists and Tuples are similar sequence types of Python, yet they are two different data types. Reason(R): List sequences are immutable and Tuple sequences are mutable.
4	Assertion (A): Tuples can only be created by using parentheses. Reason (R): t = (1, 2, 3) This python statement will create a tuple t.
5	Assertion (A): A tuple can contain other tuples as elements. Reason (R): Tuples support nesting of data structures.

ANSWERS

1	2	3	4	5
b	a	c	d	a

SHORT ANSWERS QUESTIONS

1	T1= (12, 3, 45, 'Hockey', Anil', ('a', 'b')) a. Display the first element of T1 b. Display the last element of T1 c Display T1 in reverse order. d. Display 'Anil' from tuple T1 e. Display 'b' from tuple T1
2	Write the output of the following T1 = (23, 32, 4, 5, 2, 12, 23, 7, 9, 10, 23) print(len(T1)+T1[-1]) print(T1[T1.count(23)+ len(T1) -5]) print(T1.count(T1[6])) print(T1.count(max(T1)))
3	Write a program to store the detail (Roll number, Name, Section) of three students in a nested tuple. Accept all data from the user. SAMPLE OUTPUT : ((1,'Amita','B'), (2, 'Sunita','C'), (3, 'Daya', 'A'))
4	Write the output of the following T1 = (45, 67, 98) T2 = ((45, 67, 98)) print(T1 in T2) print(45 in T2) print(45 in T1) print(T1 + T2)
5	Consider the given tuple and write the output of given statements: T1= (1, 23, 4, 5, "A",["C","D"], (23, 45), 45) 1. print(len(T1)) 2. print(T1.index(45)) 3. print(T1.count(45)) 4. print(T1 [5][0]*2) 5. print(T1 [:5]) 6. print(T1 [5:]) 7. print(T1.index("A")) 8. print(T1 [-1 : -7 : -2]) 9. print(T1 [4] + T1[5]) 10. print(max(T1))
6	Write a program in Python to find the maximum and minimum number from the given tuple.

7	Write a program in Python to find the mean of numeric values stored in the given tuple.
ANSWERS	
1	a. print(T1[0]) b. print(T1[-1]) c. print(T1[: : -1]) d. print(T1[4]) or print(T1[-2]) e. print(T1[-1][-1]) or print(T1 [5] [1])
2	34 10 3 1

3	`Det = ()`
	`tmp=()`
	`for i in range(3):`
	`  rn = int(input("Enter Roll number :"))`
	`  nm = input("Enter name of student :")`
	`  sec = input("Enter Section of student :")`
	`  tmp = (rn, nm, sec)`
	`  Det=Det + (tmp,)`
	`print(Det)`

4	False True True (45, 67, 98, 45, 67, 98)
5	1. 8 2. 7 3. 1 4. CC 5. (1, 23, 4, 5, 'A') 6. (['C','D'], (23, 45), 45) 7. 4 8. (45,['C','D'], 5) 9. Error 10. Error

6	`T = (10,2,30,4,8,5,45)`
	`print(T)`
	`minimum = min(T)`
	`maximum = max(T)`
	`print("Minimum Value : ", minimum)`
	`print("Minimum Value : ", maximum)`

7	`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)`

LONG ANSWER QUESTIONS

1	Write a program to find sum of even and odd elements of tuple
2	What is unpacking Tuple? Explain with example
3	Write a program to create a tuple and find sum of its alternate elements?
4	Write a program to find sum of elements of tuple without using sum() function?
5	Create the following tuples using a for loop containing the squares of the integers 1 through 50.
6	Write a program in Python to find the index of the given number from a tuple using linear search.
7	Write a program in Python to count the frequency of all elements of a tuple.

ANSWERS

1

```
T = eval(input("Enter value of Tuple"))
sumE = 0
sumO = 0
for x in T:
    if (x%2 == 0):
        sumE = sumE + x
    else:
        sumO = sumO + x
print(T)
print("sum of Even numbers :",sumE)
print("Sum of Odd Numbers : ", sumO)
```

2

It allows a tuple of variables on the left side of the assignment operator to be assigned respective values from a tuple on the right side. The number of variables on the left should be same as the number of elements in the tuple otherwise Python will raise a ValueError.

```
# Tuple with 3 elements
student = ("Ravi", 15, "Class 10")

# Unpacking the tuple
name, age, grade = student

# Print the variables
print("Name:", name)
print("Age:", age)
print("Grade:", grade)
```

3

```
T = (10,23,30,65,70)
sum = 0
for a in T[0:5:2]:
    sum = sum + a
print(sum)
```

4

```
T = (10,20,30)
sum = 0
for x in T:
    sum = sum + x
print(T)
print(sum)
```

5	```
	lst=[]
	for i in range(1,51):
	    lst.append(i*2)
	T=tuple(lst)
	print(T)
	``

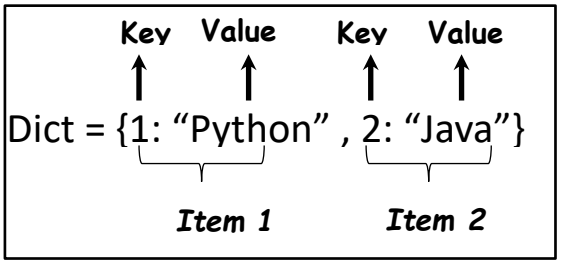
6	```
	T= (10, 2, 30, 4, 8, 5, 45)
	print("Tuple is : ",T)
	c='Y'
	while c=='Y' or c=='y':
	    n=int(input("Enter the number to be search in the tuple: "))
	    found=False
	    for i in T:
	        if i==n:
	            print("Element found at index: ",T.index(i))
	            found=True
	    if found==False:
	        print("Number is not found in the list")
	    c=input("Do you want to search more item (Y/N): ")
	``

7	```
	T= (10, 20,15, 30, 20, 40,15, 30, 20,10, 50)
	print("Tuple is: ",T)
	TF= ()
	for i in T:
	    if i not in TF:
	        TF = TF + (i,)
	for i in TF:
	    print(i,":",T.count(i))
	``

DICTIONARY

A Python dictionary is a data structure that stores the value in **key: value** pairs (called items). Values in a dictionary can be of any data type and can be duplicated, whereas keys can't be repeated and must be *immutable*.
We can also refer to a dictionary as a mapping between a set of keys and a set of values. Each key-value pair in a Dictionary is separated by a colon (:).

To create a dictionary, you need to include key: values pairs in curly braces as per following syntax:
<dictionary-name> = {<key>:<value>, <key>:<value>,…….}



✓ *Ordered: Key-value pairs have defined ordered and cannot be changed*
✓ *Mutable: Dictionaries are mutable but keys are immutable*
✗ *Duplicates: Do not allow duplicates Items with same keys*

CHARACTERISTICS OF PYTHON DICTIONARY

Unique Keys: Each key within a dictionary must be unique. If the same key is used more than once during assignment, Python will update the existing key and not add new key:value pair.
Immutable Keys: Keys must be of an immutable data type, such as strings, numbers, or tuples. Lists are not permissible as keys because they are mutable.
Case Sensitive: Dictionaries are case-sensitive, meaning that 'key' and 'Key' would be treated as distinct keys.
Key-Value Pairs: Dictionaries store data in key-value pairs, where each key is associated with a specific value.
Dictionaries are Mutable: Like list, dictionaries are also mutable. We can change the value of a certain key "in place" using the assignment statement as per syntax:
<dictionary> [key]= <value>
For example:

```
>>> Employee = {'name': 'Deepak', 'Salary': 10000, 'age': 38}
>>> print(Employee)
    {'name': 'Deepak', 'Salary': 10000, 'age': 38}
>>>
>>> Employee['Salary'] = 15000
>>> print(Employee)
    {'name': 'Deepak', 'Salary': 15000, 'age': 38}
```

MULTIPLE WAY TO CREATING DICTIONARIES

INITIALIZING A DICTIONARY

```
>>> Employee ={'name': 'Deepak', 'salary':10000, 'age':38}
>>> print(Employee)
{'name': 'Deepak', 'salary': 10000, 'age': 38}
```

CREATING AN EMPTY DICTIONARY

(a) by empty curly braces –

```
>>> Employee ={}
>>> print(Employee)
{}
```

(b) by using dict( ) method

```
>>> Employee =dict()
>>> print("Empty Dictionary: \n ", Employee)
Empty Dictionary:
 {}
```

CREATING A DICTIONARY WITH INTEGER KEYS

```
>>> Employee={1:'John', 2: 'Aryan',3: 'Raj'}
>>> print("Dictionary with the use of Integer Keys :\n", Employee)
Dictionary with the use of Integer Keys :
{1: 'John', 2: 'Aryan', 3: 'Raj'}
```

CREATING A DICTIONARY WITH MIXED KEYS

```
>>> Employee= {'Name': 'Govind', 1:[10,11,12,13]}
>>> print('\n Dictionary with use of Mixed keys: \n ', Employee)

 Dictionary with use of Mixed keys:
 {'Name': 'Govind', 1: [10, 11, 12, 13]}
```

CREATING A DICTIONARY WITH DICT( ) METHOD

(a) Specify key : Value pairs as keyword argument

```
>>> Employee=dict(name="Deepak", Salary=10000, age=38)
>>> print("\n",Employee)

 {'name': 'Deepak', 'Salary': 10000, 'age': 38}
```

(b) Specify comma-separated key: value pairs (enclose them in curly braces)

```
>>> Employee=dict({'name':"Deepak", 'Salary':10000, 'age':38})
>>> print(Employee)
{'name': 'Deepak', 'Salary': 10000, 'age': 38}
```

(c)Specify keys separately and corresponding values separately – In this method, the keys and values are enclosed separately in parentheses (i.e. as tuples) and arguments to **zip ( )**

```
>>> Employee=dict(zip(('name','salary','age'), ('Deepak', 10000,38)))
>>> print(Employee)
{'name': 'Deepak', 'salary': 10000, 'age': 38}
```

ADDING ELEMENTS ONE AT A TIME

```
>>> Dict1={}
>>> Dict1[0]='Govind'
>>> Dict1[1]='Prasad'
>>> Dict1[2]='Arya'
>>> print("\n Dictionary after adding 3 elements :-", Dict1)

 Dictionary after adding 3 elements :- {0: 'Govind', 1: 'Prasad', 2: 'Arya'}
```

ADDING SET OF VALUES TO A SINGLE KEY

```
>>> Dict1={0: 'Govind', 1: 'Prasad', 2: 'Arya'}
>>> Dict1['V']=1,2
>>> print("\n Dictionary after adding set of values:",Dict1)

 Dictionary after adding set of values: {0: 'Govind', 1: 'Prasad', 2: 'Arya', 'V': (1, 2)}
```

```
>>> Employee={'name':"Deepak", 'Salary':10000, 'age':38}
>>> Employee['dept']='Sales'
>>> print(Employee)
{'name': 'Deepak', 'Salary': 10000, 'age': 38, 'dept': 'Sales'}
```

UPDATING /MODIFYING EXISTING KEY'S VALUE

```
>>> Dict1={0: 'Govind', 1: 'Prasad', 2: 'Arya', 'V': (1, 2)}
>>> Dict1['V']=3,4
>>> print('\n Updated Dictionary: ', Dict1)

 Updated Dictionary:  {0: 'Govind', 1: 'Prasad', 2: 'Arya', 'V': (3, 4)}
```

Example 2:-

```
>>> Employee={'name':"Deepak", 'Salary':10000, 'age':38}
>>> Employee['Salary']=20000
>>> print(Employee)
{'name': 'Deepak', 'Salary': 20000, 'age': 38}
```

Note:- In dictionaries, the updation and addition of elements are similar in syntax, But for addition, the key must not exist in the dictionary and for updation , the key must exist in the dictionary.

D = {1:'AAA', 2:'BBB', 3:'CCC'}

ACCESSING ITEMS IN A DICTIONARY USING KEYS

```
>>> D={1:"AAA", 2: "BBB", 3:"CCC"}
>>> for i in D:
...     print(i, end="")
...
...

123
```

```
>>> D={1:"AAA", 2: "BBB", 3:"CCC"}
>>> for i in D:
...     print(D[i], end="")
...

AAABBBCCC
```

```
>>> D={1:"AAA", 2: "BBB", 3:"CCC"}
>>> for i in D.keys():
...     print(i, end="")
...
...

123
```

All values in the dictionary using values() method:"

```
>>> D={1:"AAA", 2: "BBB", 3:"CCC"}
>>> for i in D.values():
...     print(i, end="")
...
...

AAABBBCCC
```

All keys and values in the dictionary using items( ) method:"

```
>>> D={1:"AAA", 2: "BBB", 3:"CCC"}
>>> for k,v in D.items():
...     print(k,v,end="")
...
...

1 AAA2 BBB3 CCC
```

BUILT-IN FUNCTIONS IN DICTIONARY

len( ) : To get the length of the dictionary i.e. the count of the key: value pair, you can use the len( ) function. This function returns the length of the Dictionary. i.e. the count of elements (key:value pairs) in the dictionary.

syntax:

len (<dictionary >)

```
>>> Employee={'name':"Deepak", 'Salary':10000, 'age':38}
>>> print('Length', '=', len(Employee))
Length = 3
```

dict( ): The dict() method is used to create a dictionary in Python.

```
>>> Employee =dict()
>>> print("Empty Dictionary: \n ", Employee)
Empty Dictionary:
 {}
```

```
>>> Employee=dict(name="Deepak", Salary=10000, age=38)
>>> print("\n",Employee)

 {'name': 'Deepak', 'Salary': 10000, 'age': 38}
```

keys ( ): -This method returns all of the keys in the dictionary as a sequence of keys (in form of list). Note that these are returned in no particular order
The syntax to use this method is given below: -
<dictionary>.keys()

```
>>> Employee={'name':"Deepak", 'Salary':10000, 'age':38}
>>> print(Employee.keys())
    dict_keys(['name', 'Salary', 'age'])
```

values( )-This method returns all the values from the dictionary as a sequence (a lists) . Note that these are returned in no particular order. The syntax to use this method is given below:

<dictionary>.values()

```
>>> Employee=dict(name="Deepak", Salary=10000, age=38)
>>> print(Employee.values())
    dict_values(['Deepak', 10000, 38])
```

items( ):- This method returns all of the items in the dictionary as a sequence of (key, value) tuples . Note that these are returned in no particular order.

Syntax :-

<dictionary>.items()

```
>>> Employee=dict(name="Deepak", Salary=10000, age=38)
>>> for k,v in Employee.items():
...     print(k,v,end="#")
...
...
    name Deepak#Salary 10000#age 38#
```

get ( ) : With this method, you can get the item with the given key, similar to dictionary[key]. If the key is not present, Python by default gives error, but you can specify your own message through default argument as per following syntax:

syntax: -

dictionary >.get(key, [default])

```
>>> Employee={'name':"Deepak", 'Salary':10000, 'age':38}
>>> print(Employee.get('Salary'))
    10000
```

update( ):- This method merges key:value pairs from the new dictionary into the original dictionary, adding or replacing as needed . The items in the new dictionary are added to the old one and override any items already there with same keys :-

```
>>> Employee = {'name': 'John', 'Salary': 10000, 'age': 24}
>>> Employee.update({'dept': 'Sales'})    #non-existing key
>>> print(Employee)
    {'name': 'John', 'Salary': 10000, 'age': 24, 'dept': 'Sales'}
>>> Employee.update({'Salary': 15000}) # existing key
>>> print(Employee)
    {'name': 'John', 'Salary': 15000, 'age': 24, 'dept': 'Sales'}
```

clear() – This method removes all items from the dictionary and the dictionary becomes empty dictionary post this method .

```
>>> Employee= {'name': 'John', 'Salary':10000, 'age':24, 'dept' :'Sales'}
>>> Employee.clear()
>>> print(Employee)
    {}
```

fromkeys( ) : fromkeys() is a dictionary method used to create a new dictionary using

- A list of **keys**
- An optional **single value** that is assigned to all keys

Syntax : dict.fromkeys(<list of keys>, [optional Value])

```
>>> keys = ['name', 'age', 'dept']
>>> new_dict = dict.fromkeys(keys)
>>> print(new_dict)
    {'name': None, 'age': None, 'dept': None}
```

If no value is assigned None is assigned as value by default.

copy() :

Case 1: Creating copy using assignment operator - When you assign a dictionary name to another name using assignment operators =, it does not create a copy internally, rather it will just make two labels reference the same dictionary.

```
>>> Employee={'name':"Deepak", 'Salary':10000, 'age':38, 'dept':'Sales'}
>>> Employee1=Employee.copy()
>>> print("\n Employee", Employee, "\n Employee 1=",Employee1)

 Employee {'name': 'Deepak', 'Salary': 10000, 'age': 38, 'dept': 'Sales'}
 Employee 1= {'name': 'Deepak', 'Salary': 10000, 'age': 38, 'dept': 'Sales'}
```

Case 2: Creating copy using the copy () method:

(i) If the values referenced by the keys are immutable, then any changes made in the copy CREATED with copy () will not be reflected in the original dictionary as the copied dictionary has own set of referencing keys.

```
>>> Employee={'name':"Deepak", 'Salary':10000, 'age':38}
>>> Employee1=Employee.copy()
>>> Employee1['dept']='sales'
>>> print("\n Employee Dictioanry =", Employee, "\n Employee1 Dictionary =", Employee1)

 Employee Dictioanry = {'name': 'Deepak', 'Salary': 10000, 'age': 38}
 Employee1 Dictionary = {'name': 'Deepak', 'Salary': 10000, 'age': 38, 'dept': 'sales'}
```

(ii) If the values referenced by the key are Mutable (such as list) then the keys will be referring to the same Python list object (that is the memory address of the values won't change) but list being mutable can change

```
>>> d1={1:[1,2,3], 2: [3,4,5]}
>>> d2=d1.copy()
>>> d2[1].append(4)
>>> print("\n d1 Dictionary= ", d1, "\n d2 Dictionary= ", d2)

 d1 Dictionary=  {1: [1, 2, 3, 4], 2: [3, 4, 5]}
 d2 Dictionary=  {1: [1, 2, 3, 4], 2: [3, 4, 5]}
```

pop( ) : This method removes the specified item from the dictionary and return the corresponding value.

```
>>> Employee={'name':"Deepak", 'Salary':10000, 'age':38, 'dept':'sales'}
>>> Employee.pop('dept')
'sales'
```

popitem( ) : The popitem ( ) removes and returns a (key, value) pair from the dictionary. It returns the item which was the last item entered in the dictionary.
Syntax :- <dictionary>.popitem()

```
>>> Employee={'name':"Deepak", 'Salary':10000, 'age':38, 'dept':'sales'}
>>> print(Employee.popitem())
('dept', 'sales')
```

setdefault() : This method inserts a new key: value pair ONLY IF the key does not exist and If the key already exists, it returns the current value of the key.

Syntax :- <dictionary>.setdefault (<key>, <value>)

```
>>> Employee={'name':"Deepak", 'Salary':10000, 'age':38, 'dept':'sales'}
>>> Employee.setdefault('Gender', 'Male')
'Male'
>>> print(Employee)
{'name': 'Deepak', 'Salary': 10000, 'age': 38, 'dept': 'sales', 'Gender': 'Male'}
```

max( ) : returns the maximum key

min( ) : returns the minimum key

sum( ): return the sum of keys

```
>>> d1={1:23, 2:56, 3:78, 4:37}
>>> print(min(d1),max(d1),sum(d1))
1 4 10
```

sorted( ): returns the dictionary in sorted order on keys/values

```
>>> D={1:"AAA", 2: "BBB", 3:"CCC"}
>>> print(sorted(Employee.items()))
[('Gender', 'Male'), ('Salary', 10000), ('age', 38), ('dept', 'sales'), ('name', 'Deepak')]
```

QUESTION BANK

MULTIPLE CHOICE QUESTIONS

1.	Which value is assigned to keys, if no value is specified with the fromkeys ( ) method? (a) 0 (b) 1 (c) None (d) any of these
2.	What does the fromkeys( ) method return? (a) A list (b) A string (c) A dictionary (d)A tuple
3.	What will be the output? d1 = {"ABC ": 40 , "XYZ":45 } d2= {"ABC" :46 , "XYZ" :45} d1==d2 (a) True (b) False (c) None (d) Error
4.	What is the output of the following code ? a = {1 : "A", 2: "B", 3: "C" } b=a.copy( ) b[2]= "D" print(a) (a) Error , copy ( ) method doesn't exist for dictionaries (b) {1 : 'A' , 2 : 'B', 3 : 'C'} (c) {1 : 'A' , 2 : 'D', 3 : 'C'} (d) "None" is printed
5.	Which method is used to add new key value pair if not exists ? (a) get( ) (b) insert() (c) add( ) (d) setdefault()
6.	Which method is used to remove a key-value pair from a dictionary? (a) pop( ) (b) remove( ) (c) delete( ) (d) clear()
7.	What will be the output of the following Python code snippet? my_dict = {'a': 1, 'b': 2, 'c': 3} result = my_dict.get('b', 0) print(result) (a) 1 (b) 2 (c) 3 (d) 0
8.	What will be the output of the following Python code snippet? my_dict = {'a': 1, 'b': 2, 'c': 3} result = my_dict.items() print(result) a) [('a', 1), ('b', 2), ('c', 3)] b) (('a', 1), ('b', 2), ('c', 3)) c) dict_items([('a', 1), ('b', 2), ('c', 3)]) d) {('a', 1), ('b', 2), ('c', 3)}
9.	Which of the following statements about dictionaries in Python is true? a) Dictionaries are ordered collections b) Dictionaries can contain duplicate keys c) Dictionaries are indexed by integers d) Dictionaries are immutable

10	What will be the output of the following code ?
	d = {'a': 100, 'b': 200, 'c': 300}
	print(sum(d.values()) // len(d.keys()))
	a)200 b) 600 c) 100.0 d) TypeError

ANSWER

1	2	3	4	5	6	7	8	9	10
C	C	B	B	D	A	B	C	A	A

ASSERTION AND REASONING QUESTION

In the following question, 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 the correct explanation of A.
(b) Both A and R are true but R is not the correct explanation of A.
(c) A is true but R is false
(d) A is false but R is true.

1. Assertion: Dictionaries are similar to lists, in a way
   Reason: Dictionaries are similar to lists, only in mutability.
2. Assertion: Dictionaries use indexes exactly like lists and tuples, to access elements.
   Reason: Dictionary elements are in the form of key: Value pairs where key is used to access corresponding value.
3. Assertion: Existence of a key in a dictionary can be ascertained with the membership operators.
   Reason: The membership operators in and not in only work with the dictionary's keys and not with values.
4. Assertion: Immutable keys of a dictionary can be duplicated in a dictionary
   Reason: A dictionary's key must be unique
5. Assertion: Mutable type dictionaries internally store elements through immutable keys.
   Reason: In every **key:value** pair, the **key** must be of immutable type always to facilitate internal mapping of elements.
6. Assertion: The fromkeys() method creates a new dictionary with the specified keys.
   Reason: If no value is given, it assigns None to all keys.
7. Assertion**:** In a dictionary, values must be unique.
   Reason**:** Dictionary keys must not be repeated.
8. Assertion: Dictionary is a collection of key-value pairs.
   Reasoning: Each key in a dictionary map to a corresponding value. Keys are unique and act as the index.
9. Assertion: clear( ) method removes all elements from the dictionary.
   Reasoning: len( )function cannot be used to find the length of a dictionary.
10. Assertion**:** Dictionaries in Python preserve insertion order.
    Reason**:** Items in a dictionary appear in the same order they were added.

ANSWER

1	2	3	4	5	6	7	8	9	10
A	D	A	D	A	B	D	B	C	A

	VERY SHORT ANSWER QUESTIONS
1.	What is the syntax to create an empty dictionary?
2.	How do you add a key-value pair to a dictionary?
3.	How do you access a value in a dictionary?
4.	What happens if you try to access a non-existent key?
5.	How do you remove a key-value pair from a dictionary?
6.	What is the output of dict.keys()?
7.	How do you merge two dictionaries?

8.	What is the output of dict.get(key, default)?
9.	How do you iterate over key-value pairs in a dictionary?
10.	What is the output of dict.clear()?

ANSWERS

1. {} or dict()
2. dict[key] = value
3. dict[key]
4. KeyError
5. del dict[key] or dict.pop(key)
6. A list of keys
7. dict1.update(dict2)
8. The value if key exists, otherwise default
9. for key, value in dict.items():
10. An empty dictionary

SHORT ANSWER QUESTIONS

1. Why are dictionaries called mutable types?
2. How are individual elements of dictionaries accessed?
3. What will be the output of the following code:

```
d= {“x”:10, “y”:15, “z”:20, “w”:5}
lst= [“x”, “w”]
s=0
for k, v in d.items ():
    if k in lst :
        s=s+d[k]
print(s)
```

4. What will be the output of the following commands (i) and (ii).

```
key = 11
numbers = {1:2, 3:4, 5:6, 7:8}

(i) del numbers[key]
(ii) numbers.pop(key, -1)
```

5. Can you remove key: value pairs from a dictionary and if so, how?
6. How are dictionary different from lists.
7. How is indexing of a dictionary different from that of a list or a string?
8. What will be the result of the following code?

```
d = {"Jo":1,"Ra":2}
d.update({"Phoebe":2})
print(dict)
```

9. What will be the output of following Python code?

```
d1 = {"a":10,"b":2,"c":3}
str1=""
for i in d1:
  str1 = str1 + str(d1[i]) + " "
  str2 = str1[:-1]
print(str2[::-1])
```

10. Write a python program to get values of a dictionary in ascending order.

For Example:

Original Dictionary = {1:25, 2:21, 3:23}

Expected Output = [21, 25, 32]

ANSWERS

Ans 1 - Dictionaries can be changed by adding new key-value pairs and by deleting or changing the existing ones. Hence, they are called mutable types.
For example:

```
d = {"a": 1, "b": 2}
d["c"] = 3
d["b"] = 4
del d["a"]
print(d)
```

OUTPUT : {'b': 4, 'c': 3}

Ans 2 - Individual elements of a dictionary can be accessed by using their corresponding keys as per the syntax shown below:

```
<dictionary-name> [<key>]
For example:
d = {'a': 1, 'b': 2, 'c': 3}
print(d['a'])
```

Ans 3 – 15

Ans 4 - The statement (i) del numbers[key] will throw an Error since the given key (11) is not in the dictionary.
The statement (ii) will display -1 as the key 11 is not in the dictionary. It won't produce any error.

Ans 5 - Yes, key:value pairs can be removed from a dictionary. The different methods to remove key:value pairs are given below:

By using del command:

It is used to delete an item with the specified key name. The syntax for doing so is as given below:

```
del <dictionary>[<key>]

For example:

dict = {'list': 'mutable', 'tuple': 'immutable', 'dictionary': 'mutable'}
del dict["tuple"]
print(dict)
```

Ans 6:- **Structure**: A list holds individual values; a dictionary holds key-value pairs.
Access Method: List elements are accessed by index; dictionary elements are accessed by key.

Ans 7 :- In lists or strings, the elements are accessed through their index where as in dictionaries, the elements are accessed through the keys defined in the key:value pairs. Moreover, lists and strings are ordered set of elements but dictionaries are unordered set of elements so its elements cannot be accessed as per specific order.

Ans 8:-

{'Jo': 1, 'Ra': 2, 'Phoebe': 2}

Reason — The update() method updates the dictionary with the elements from another dictionary object or from an iterable of key/value pairs.
Here, {"Phoebe":2} is a dictionary of key named "Phoebe" which is not present in dict.
Therefore, update function will add this key and its corresponding value to original dictionary dict.

Ans 9:- 3 2 01

Ans 10:-

```
d={1:21, 2:32, 3:25}
print(sorted(d.values()))
print(d)
```

LONG ANSWER TYPE QUESTIONS

Q 1:- Write a Python program to create a dictionary that stores the names of students as keys and their respective marks as values. The program should also calculate the average marks of all students.

Q 2:- Write a program to 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 .

Q 3:-Consider the following code and answer the questions that follow :-

Book = {1: "Thriller" , 2: "Mystery " , 3: "Crime ", 4: "Children Stories"}

Library = {5 : "Madras Diaries ", 6 : "Malgudi Days "}

(i) Ramesh needs to change the title in the dictionary Book from 'Crime' to 'Crime Thriller'. He has written the following command. Book ['Crime'] = 'Crime Thriller' But he is not getting the answer. Help him choose the correct command:

(a) Book [2] = 'Crime Thriller' (b) Book [3] = 'Crime Thriller'

(c) Book [2] =('Crime Thriller') (d) Book [3]= ('Crime Thriller')

(ii) The command to merge the dictionary Book into Library the command would be

(a) d= Book +Library (b) print (Book +Library)

(c) Book.update (Library) (d) Library.update(Book)

(iii) What will be the output of the following line of code : print(list(Library))

(a) ['5' , 'Madras Diaries , '6', 'malgudi Days']

(b) ('5' , 'Madras Diaries , '6', 'malgudi Days')

(c) ['Madras Diaries' 'Malgudi Days']

(d) ['5', '6']

(iv) In order to check whether the key 2 is present in the dictionary Book. Ramesh uses the following command : 2 in Book

He gets the answer "True". Now to check whether the name 'Madras Diaries' exists in the dictionary Library, he uses the following command: 'Madras Diaries' in Library But he gets the answer as 'False'. Select the correct reason as to why is he getting 'False' as an answer.

(a) We cannot use **in** operator with values. In operator works with keys only.

(b) We must use the function Library. values () along with the in operator

(c) We must use the Library.items ( ) function instead of the in operator.

(d) Both b and c above are correct.

(v) With reference to the above declared dictionaries, predict the output of the following code fragments: -

Code 1	**Code 2**
Library = Book Library.pop (2) print (Library) print(Book)	Library=Book.copy () Library.pop(2) print(Library) print(Book)

Q 4:- Consider the following dictionary stateCapital :

stateCapital = {"AndhraPradesh" :"Hyderabad", "Bihar": "Patna", "Maharashtra":"Mumbai", "Rajasthan":"Jaipur"}

Based on above dictionary find the output of the following statements :

(a) print(stateCapital.get("Bihar"))

(b) print(stateCapital.keys())

(c) print(stateCapital.values())

(d) print(stateCapital.items())

(e) print(len(stateCapital))

(f) print("Maharashtra" in stateCapital)
(g) print(stateCapital.get("Assam"))
(h) del stateCapital["Andhra Pradesh"]

Q 5:- Write a Python program to print a dictionary where the keys are numbers between 1 and n (both included) and the values are square of keys

Q 6:- A dictionary in Python is an ordered collection of data values that stores key:value pairs of values instead of a single value as an element. Multiple key:value pairs are separated by commas and all of these are enclosed in { }. An empty dictionary without any items is written with just two curly braces, i.e. as: {}. Consider the dictionary D1 created as :

D1 = {"Anu " :25 , "Ritu " :30 , "Dia " :27}

(a) Which of the following methods returns a sorted sequence of the keys in the list?
(i) sorted () (ii) fromkeys() (iii) update() (iv) items()

(b) Predict the output (considering dictionary D1 created above):-
print((D1 ["Anu"]+ D1["Anu "]))
(i) 25 (ii) 30 (iii) 55 (iv) 50

(c) Which of the following is an invalid statement (considering dictionary D1 created above)?
(i) D1["Anu"]=33 (ii) D1["Ritu']= 46 (iii) D1["Dia"]=44 (iv) D1[33]= "Anu"

(d) The clear( ) method is used to remove the elements of the dictionary. It can also be used to delete a particular key value pair in a dictionary – True / False

(e) Which of the following statements will create a dictionary?
(i) d = { } (ii) d= {"Dan" :40 , "Don": 45}
(iii) d = {40: "Ron", 45: "Rob"} (iv) All of these

Q 7:- Write a program to create a dictionary namely Dict with 10 keys 0…9 , each having values as 200. Update the first and last values by adding 200 to each of them.

Q 8:-Why is the following code producing error ? Correct the error and display the output of the corrected code .

```
box, jars, crates = {}, {}, {}
box['biscuit']=1
box ['cake']=3
jars['jam']=4
crates['box']=box
crates['jars']=jars
print(len(crates[box]))
```

Q. 9 :- Find the output of the following code:-

(a)
```
fruit= {}
L1= ['Apple', 'banana', 'apple']
for index in L1 :
   if index in fruit:
      fruit[index]+=1
   else:
      fruit[index]=1
print(len(fruit), "\n", fruit)
```

(b)
```
d1= {5: 'number', 'a': 'string', (1,2): 'tuple'}
print( 'Dictionary contents ')
for x in d1.keys():
   print(x, ":", d1[x], end= "")
   print(d1[x]*3)
   print( )
```

Q.10 :- Write a program to input your friends names and their Phone Numbers and store them in the dictionary as the key value pair. Perform the following operations on the dictionary :-

(a) Display the name and phone number of all your friends.
(b) Add a new key- value pair in this dictionary and display the modified dictionary.
(c) Delete a particular friend from the dictionary
(d) Modify the phone number of an existing friend
(e) Check if a friend is present in the dictionary or not
(f) Display the names of all friends in sorted order.

ANSWERS

Ans 1:-

```
student_marks = {}
    # Get the number of students
num_students = int(input("Enter the number of students: "))

    # Collect student names and marks
for _ in range(num_students):
    name = input("Enter student name: ")
    marks = float(input(f"Enter marks for {name}: "))
    student_marks[name] = marks

    # Calculate average marks
if student_marks:
    average_marks = sum(student_marks.values()) / len(student_marks)
    print("\nStudent Grades:")
    for student, marks in student_marks.items():
        print(f"{student}: {marks}")
    print(f"\nAverage Grade: {average_marks:.2f}")
else:
    print("No student data provided.")
```

Ans 2:-

```
n= int (input("How many Students"))
stu= { }
for i in range (1, n+1):
       print ("Enter details of student", (i))
       rollno =int(input("Roll number :"))
       name= input("Name : ")
       marks= float (input("Marks :  "))
       d= { "Roll_no " : rollno, "Name " : name , "Marks": marks}
       key= "Stu" + str (i)
       stu [key]=d
print ("Student with marks > 75 are: ")
for i  in range (1, n+1):
        key= "Stu" +str (i)
        if stu [key]["Marks"] >= 75 :
                print(stu [key])
```

Ans 3:-

(i) b (ii) d (iii) d (iv) a
(v) **Code 1 :**

{1: “Thriller”, 3: “Crime ”, 4: “Children Stories ”}
{1: “Thriller”, 3: “Crime ”, 4: “Children Stories ”}

Code 2:-

{1: “Thriller”, 3: “Crime ”, 4: “Children Stories ”}
{1: “Thriller”, 2: “Mystery ”, 3: “Crime , 4 : “Children Stories” }

Ans 4:-

(a) Patna
(b) dict_keys(['AndhraPradesh', 'Bihar', 'Maharashtra', 'Rajasthan'])
(c) dict_values(['Hyderabad', 'Patna', 'Mumbai', 'Jaipur'])
(d) dict_items([('AndhraPradesh', 'Hyderabad'), ('Bihar', 'Patna'), ('Maharashtra', 'Mumbai'), ('Rajasthan', 'Jaipur')])
(e) 4
(f) True
(g) None
(h) It will give KeyError as there is no key as “Andhra Pradesh’(space in between)

Ans 5:-

```
l=int(input("Enter the Limit : "))
d = dict()
for x in range(1,l+1):
   d[x]=x**2
print(d)
```

Ans 6:-

(a) (i) sorted (b) iv) 50 (c) (ii) D1[“Ritu’]= 46 (d) False (e) (iv) All of these

Ans 7:-

```
>>> Dct=dict.fromkeys(range(10),200)
>>> Dct[0]+=200
>>> Dct[9]+=200
>>> print(Dct)
{0: 400, 1: 200, 2: 200, 3: 200, 4: 200, 5: 200, 6: 200, 7: 200, 8: 200, 9: 400}
```

Ans 8:-

The last line of the code is producing error because of the expression crates[box]. This is because there is a dictionary box defined above and it is assigned to crates [‘box’] (notice the string ‘box’). But the above is trying to access the crates dictionary using the box dictionary whereas it should be accessed using the ‘box’ as the key is the ‘box’ string and not the box dictionary.

So, the corrected code will have the last line replaced as :-
print(len(crates[‘box’]))
and it will produce the output as :-
2 (which the length of crates [‘box’] value which is the box dictionary.)

Ans 9:-

(a) 3
{'Apple': 1, 'banana': 1, 'apple': 1}

(b) Dictionary contents
5 : numbernumbernumbernumber
a : stringstringstringstring
(1, 2) : tupletupletupletuple

Ans 10 :-

```
n=int(input("How many friends"))
fd={}
for i in range(n):
   print("Enter details of friends ", (i+1))
   name=input("Name : ")
   ph=int(input("Phone: "))
   fd[name]=ph
   print("Friendd dictioanry is ", fd)
ch=0
while ch!=7:
   print("\t Menu")
   print("1. Display all friends " )
   print("2. Add new friend ")
   print("3. Delete a friend ")
   print("4. Modify a phone number ")
   print("5. Search for a friend " )
   print("6. Sort on names ")
   print("7. Exit")
   ch=int(input("Enter your choices (1...7 )"))
   if ch==1:
      print(fd)
   elif ch==2:
      print("Enter details of new friend ")
      name =input("Name = ")
      ph=int(input("Phone : "))
      fd[name]=ph
   elif ch==3:
      nm=input("Friend Name to be deleted ")
      res=fd.pop(nm,-1)
      if res != -1:
         print(res, "deleted")
      else:
         print("No such friend ")
   elif ch==4:
      name =input("friend Name : ")
      ph=int (input("Changed Phone : "))
      fd[name]=ph
   elif ch==5:
      name =input("Friend Name : ")
      if name in fd:
         print(name , "exists in the dictonary ")
   else:
      print(name , "does not exist in the dictionary .")
 elif ch==6:
   lst=sorted(fd)
   print("{", end="")
   for a in lst :
      print(a, ":", fd[a], end ="")
   print("}")
 elif ch==7:
   break
 else:
   print("Valid Choices are 1....7")
```

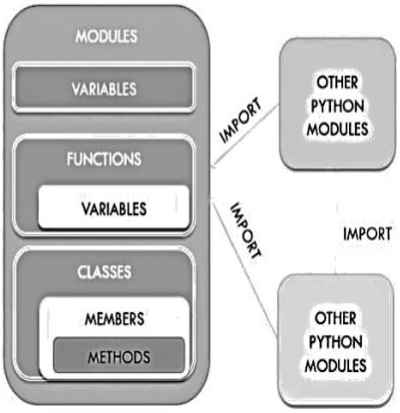
	PRACTICE QUESTIONS
1.	Write a Python program to count the frequency of each word(irrespective of case) in a given string and store the result in a dictionary. e.g. Input: "Hello world, hello python" Output: {"hello": 2, "world": 1, "python": 1}
2.	Create a dictionary to store student information, including name, age, and grade. Write a program to add a new student also write code to retrieve a student's information by name.
3.	Given a dictionary of student grades, write a program to calculate the average grade for each student and return the result in a new dictionary. Input: {"John": [80, 70, 90], "Jane": [70, 80, 70]} Output: {"John": 80.0, "Jane": 73.33}
4.	Write a program to merge two dictionaries and return the result. Input: dict1 = {"a": 1, "b": 2}, dict2 = {"a": 3, "d": 4} Output: {"a": 3, "b": 2, "d": 4}
5.	Given a dictionary of employee data, write a program to find the highest salary and return the employee's name and salary. Input: {"John": 50000, "Jane": 60000, "Bob": 70000} Output: ("Bob", 70000)
6.	Develop a Python program that utilizes a dictionary to store the details of employees in a company, including their names, ages, and salaries. The program should also calculate the total salary expenditure
7.	Design a Python program that employs a dictionary to store the details of books in a library, including their titles, authors, and publication years. The program should also find books by a specific author.
8.	Write a Python program that utilizes a dictionary to store the details of customers in an ecommerce platform, including their names, addresses, and order histories. The program should find the customer with the highest total order value.
9.	Design a Python program that employs a dictionary to store the details of patients in a hospital, including their names, ages, and medical records. The program should find patients by a specific age range.
10.	Write a Python program that utilizes a dictionary to store the details of products in an inventory management system, including their names, quantities, and prices. The program should update the quantity of a product as per users need.
11.	Develop a Python program that uses a dictionary to store the details of flights in an airline management system, including their unique number, source, destination, departure times, and arrival times. The program should find flights based on its number or source or destination as per users need. *(Hint : Menu Driven Program)*

INTRODUCTION TO PYTHON MODULES

INTRODUCTION

- A Library refers to a collection of modules that together cater to specific type of needs or applications.
- A module is simply a python file where statements, classes, Objects, functions, constants and variables are defined. The file name is the module name with .py extension.
- A module, in general:
  - is independent grouping of code and data (variables, definitions , statements and functions).
  - Can be re-used in other programs.
  - Can depend on other modules.

Composition/Structure of python module



IMPORTING MODULES IN A PYTHON PROGRAM

In Python, the **import** statement allows access to modules and their definitions, such as functions, classes, and variables, supporting modular and efficient code organization. **import** statement allows a program to access code from external modules, promoting modularity and reuse.

(i) To import entire module : the ***import <module>*** command

(ii) To import specific objects from a module : ***from <module> import <object>*** command

A. **Importing Entire module:-** The import statement can be used to import an entire module and even for importing selected items.To import an entire module, the import statement can be used as per the following syntax:-

import modulename1, [modulename 2,....]

Example :-

```
import  math
value = math.sqrt (25)
print(value)
```

B. **Importing specific Object from a Module :-** if you want to import some specific items not all from a module, then you can use from <module> import statement as per the following syntax :-

from <module> import <objectname>, [, <objectname> [.....]|*

(a) **To import single Object** – if you want to import a single object from the module like this so that you do not have to prefix the module's name , you can write the name of object after keyword import.

from <module> import <objectname>

#example :-

from math import pi

print(pi)

Note :- in this method the constant pi can be used directly and you need not prefix it with module name like print(math.pi).

(b) **To import Multiple object -** if you want to import multiple object from the module like this so that you don't have to prefix the module's name , you can write the comma separated list of object after the keyword import

from <module> import <objectname1>,<objectname2>

#example:

from math import sqrt, pow

(c) **To import All object of a module :-** if you want to import all the items from the module like this so that you don't have to prefix the module's name , you can write :

from <module name> import *

#example

from math import *

WORKING WITH MATH MODULE

Python's standard library provides a module namely **math** for math related functions that work with all number types except for complex numbers.

import math

Some import functions in **math** module:

Function	Description	Example	Output
sqrt( )	Returns the square root of a number	math.sqrt(49)	7.0
ceil( )	Returns the upper integer	math.ceil(81.3) math.ceil(-81.3)	82.0 -81.0
floor( )	Returns the lower integer	math.floor(81.3) math.floor(-81.3)	81.0 -82.0
pow( )	Calculate the power of a number	math.pow(2,3)	8.0
fabs( )	Returns the absolute value of a number	math.fabs(-5.6)	5.6
sin( )	Returns the sine of a number	math.sin(90)	0.89399
cos()	Returns the cosine of a number	math.cos(90)	-0.448073
tan()	Returns the tangent of a number	math.tan(90)	-1.995200
degrees()	The degrees () converts angle x from radians to degrees	math.degrees(3.14)	179.91
radians()	The radians ( ) convert angle x from degrees to radians.	Math.radian(179.91)	3.14

Example

Write the corresponding Python expression for the following mathematical expression-

$(i)\ \sqrt{a^2+b^2+c^2}$ $(ii)\ 2-ye^{2y}+4y$ $(iii)\ p+\frac{q}{(r+s)^4}$

$(iv)\ (\cos x/\tan x)+x$ $(v)\ |e^2-x|$

Solution :-

(i) math.sqrt(a*a+b*b*c*c)

(ii) 2-y*math.exp(2*y)+4*y

(iii) p+q/math.pow((r+s),4)

(iv) math.cos(x)/ math.tan(x))+x

(v) math.fabs (math.exp(2)-x)

WORKING WITH RANDOM MODULE

To work with the functions of a **random** module, we must import a random module in the program.

import random

Function	Description	Example	Output
random ( )	It returns a random float x, such that 0 ≤ x<1	random.random()	0.2819547
randint(a, b)	It returns an integer x between a & b such that a ≤ x ≤ b	random.randint(1,10)	5
randrange ([start,] stop [,step])	It returns a random item from the given range upto stop-1.	random.randrange(100,1000,3)	150
choice(<seq>) **beyond syllabus*	It returns a random element from the sequence	random.choice('welcome') random.choice([1,4,2,3])	'o' 2

Example

1. To generate a random floating-point number between 0.0 to 1.0 .
   import random
   print(random.random( ))
   Output:- 0.02235193431 # The output generated is between range 0.0 to 1.0
2. To generate a random floating point number between to certain numbers using random ( ):
   (a) multiply random( ) with difference of upper limit with lower limit, i.e. (upper limit-lower limit)
   (b) add to it lower limit
   import random
   print (random.random( )*(35-15)+15)
   Output:- 28.3071872734 # The output generated is float between range 15 to 35
3. To generate a random integer number in range 15 to 35 using randint ( ):
   print (random.randint(15,35))
   Output:- 16 # The output generated is integer between range 15 to 35

Using randrange( ) function

The function randrange( ) can be used in following three ways ;-

(i) random.randrange(<stopvalue>) – to generate a random number in the range 0 to <stop value>
example :- random.randrange(45)
output :- 13 *# a random number generated in the range 0….45*

(ii) random.randrange(<start>, <stop>) :- to generate a random number in the range <start value> to <stop value>
for example :- random.randrange(11,45)
output :- 25 *# a random number generated in the range 11….45*

(iii) random.randrange(<start>, <stop>, <step>) :- to generate a random number in the range <start value> to <stop value> but difference between two such generated random numbers will be a multiple of <step> value.
for example :-
random.randrange(11,45,4) will always give a value from
11,15,19,23,27,31,35,39,43

i) random.randrange(11,45,4)
output :- 15
ii) random.randrange(11,45,4)
output :- 35
iii) random.randrange(11,45,4)

 # See each generated random number is one of the above given values

output :- 39

WORKING WITH STATISTICS MODULE

To work with the functions of the statistics module, we must import the statistics module in the program.

import statistics

OR

From statistics import mean, median, mode

Sr. No	Function	Description	Example	Output
1.	mean (<seq>)	It returns mean of a list of numbers	print(statistics.mean([1, 3, 5, 7, 9, 11]))	6
2.	median(<seq>)	It returns the median (middle value) of the given data set	print(statistics.median([1, 3, 5, 7, 9, 11, 13]))	7
3.	mode(<seq>)	It returns the mode (central tendency) of the given numeric or nominal data set.	print(statistics.mode([1, 3, 3, 3, 5, 7, 7, 9]))	3

QUESTION BANK

MULTIPLE CHOICE QUESTIONS

1. What is the value returned by math.floor(3.4) ?
   (a) 3 (b) 4 (c) 4.0 (d) 3.0
2. Which of the following is not a function/method of the random module in Python ?
   (a) randfloat( ) (b) randint( ) (c) random( ) (d) randrange( )
3. What will the following code result as ?

```
import math
x=100
print(x>0 and math.sqrt(x))
```

   (a) True (b) 1 (c) 10 (d) 10.0
4. What is the output of the function shown below if the random module has already been imported ?

```
random.randint(3.5,7)
```

   (a) Error
   (b) Any integer between 3.5 and 7 including 7
   (c) Any integer between 3.5 and 7, excluding 7
   (d) The integer closest to the mean of 3.5 and 7
5. Which of the following options is the possible outcome of the function shown below ?
   random.randrange (1,100,10)
   (a) 32 (b) 67 (c) 91 (d) 80
6. How can you import a specific function from a module in Python?
   (a) using the 'import' keyword followed by the function name
   (b) using the 'import' keyword followed by the module name and function name separated by a dot
   (c) using the 'require' keyword followed by the function name
   (d) using the 'use' keyword followed by the module name and function name separated by a dot
7. Which keyword is used to create an alias while importing a module in Python?
   (a) rename (b) alias (c) as (d) with
8. The collection of modules and package that together cater to a specific type of applications or requirements, is called
   (a) classes (b) library (c) module (d) documentation

9. What will be the output of the following Python code?
   random.randrange(0,91,5)
   (a) 95 (b) 18 (c) 79 (d) 10
10. Which of the following is equivalent to random.randrange(3)?
   (a) range(3)
   (b) random.choice(range(0, 3))
   (c) random.shuffle(range(3))
   (d) andom.select(range(3))

ANSWERS

1	2	3	4	5	6	7	8	9	10
a	a	d	a	c	b	c	c	d	b

ASSERTION AND REASONING QUESTIONS

In the following question, 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 the correct explanation of A.
(b) Both A and R are true but R is not the correct explanation of A.
(c) A is true but R is false
(d) A is false but R is true.

1. **Assertion :-** After importing a modules through import statement , all its function definitions, variable , constants etc are made available in the program.
   Reason :-Imported module's definitions do not become part of the program's namespace if imported through import <module> statement.
2. **Assertion :-** If an item is imported through from <module> import <item> statement then you do not use the module name along with the imported item.
   Reason :-The from <module> import <item> command adds the imported items directly to the namespace of the program.
3. **Assertion :** To use a function from a particular module, we need to import the module.
   Reason : import statement should be written after function call to a function of that module.
4. **Assertion :** Python Standard Library consists of various modules.
   Reason : A function in a module is used to simplify the code and avoids repetition
5. **Assertion :-.** Python's built-in functions, which are part of the Python standard library, can directly be used without specifying their module name.
   Reason :- Python's standard library's built-in functions are made available by default in the namespace of a program.

ANSWERS

1	2	3	4	5
b	a	c	b	a

SHORT ANSWER QUESTIONS

1. Name the Python Library module which should be imported to use the following functions:
   (i) sin ( ) (ii) randint ( )
2. What possible output(s) are expected to be displayed on screen at the time of execution of the program from the following code? Also specify the maximum values that can be assigned to each of the variables Lower and Upper.

```
import random
AR =[20,20,40,50,60,70]
Lower= random.randint(1,3)
Upper=random.randint(2,4)
for k in range (Lower, Upper+1):
  print(AR[k] , end= '#')
```

(i) 10#40#70# (ii) 30#40#50# (iii) 50#60#70# (iv) 40#50#70#

3. What possible output(s) are expected to be displayed on screen at the time of execution of the program from the following code? Also specify the minimum and maximum values that can be assigned to the variable End.

```
import random
Colours =['VIOLET', 'INDIGO' , 'BLUE', 'GREEN', 'YELLOW', 'ORANGE', 'RED']
End = randrange(2)+3
Begin=randrange(End) +1
for i in range (Begin, End):
  print(Colours [i] , end= '&')
```

(i) INDIGO&BLUE&GREEN& (ii) VIOLET&INDIGO&BLUE&
(iii) BLUE&GREEN&YELLOW& (iv) GREEN&YELLOW&ORANGE&

4. Identify the correct possible output for the following Python code:

```
import random
for N in range (2, 5,2):
  print (random.randrange(1,N), end= '#' )
```

(a) 1#3#5 (b) 2#3# (c) 1#4# (d) 1#3#

5. The radius of a sphere is 7.5 metres. Write Python script to calculate its area and volume. (Area of a sphere = $4\pi r^2$, Volume of a sphere = $\frac{4}{3}\pi r^3$.)
6. What is a Python module? What is its significance?
7. Which import statement does not require the imported module's name along with imported variables /definitions in the program and why?
8. A triangle has three sides a,b,c, take values from user, calculate and display its areas using Heron's formula as

$$s = \frac{a+b+c}{2}$$

$$\text{Area} = \sqrt{s(s-a)(s-b)(s-c)}$$

9. Write a random number generator that random numbers between 1 and 6 (simulates a dice) until the user wants.
10. State True/false: -
    (a) A python program and a python module mean the same.
    (b) A python program and a Python module have the same .py file extension
    (c) Any folder having .py files is a python package
    (d) The Statement from <module> import <objects> is used to import a module in full.

ANSWERS

1. (i) math module (ii) random module

2. (ii) 30#40#50#
    a. Maximum value of Lower: 3

b. Maximum value of Upper: 4

3. (i) INDIGO&BLUE&GREEN&
   a. Minimum Value of End = 3
   b. Maximum Value of End = 4
4. (d) 1#3#
5.
```
import math
r= 7.5
area= 4*math.pi*r*r
volume =(4/3)*math.pi*math.pow(r,3)
print('Radius of the sphere : ' , r, 'metres')
print( 'Area of the sphere :',area , 'units square')
print( 'Volume of the sphere ' ,volume , 'units cube')
```
6. A "module" is a chunk of python code that exists in its own (.py) file and is intended to be used by Python code outside itself.
   Module allow one to bundle together code in a form in which it can easily be used later. The modules can be "imported" in other programs so the functions and other definitions in imported modules become available to code that imports them.
7. The *from <module> import<item>* statement does not require the imported module's name in the program code because it modifies the namespace of the current program and brings all imported variables /definitions in it.
8.
```
import math
a,b,c=17,23,30
s= (a+b+c)/2
area= math.sqrt(s*(s-a)*(s-b)*(s-c))
print('Sides of triangle: ', a,b,c)
print('Area: ', area, 'units square')
```
9.
```
import random
min=1
max=6
roll_again='y'
while roll_again=='y' or roll_again=='Y':
  print("Rollin the dice.....")
  val=random.randint(min, max)
  print("You get .....", val)
  roll_again=input("Roll dice again ? Y/N-  ")
```
10. (a) False (b) True (c) False (d) False

SOCIETY, LAW AND ETHICS

DIGITAL FOOTPRINT

A digital footprint is the trail of data you leave behind while using the internet or digital devices. It includes everything from your browsing history, social media posts, and online purchases to metadata like IP addresses and cookies. There are two types: **active** (data you intentionally share, like posts or comments) and **passive** (data collected without your direct input, like website tracking or location data).

Why It Matters

Privacy: Companies, governments, or hackers can track your behavior, preferences, and location.
Reputation: Employers, schools, or others may judge you based on your online activity.
Security: A large digital footprint increases risks of identity theft or targeted scams.

Managing Your Digital Footprint

- Review Privacy Settings: Limit what apps and websites can collect.
- Be Cautious with Sharing: Think before posting or sharing personal info.
- Use Strong Passwords: Protect accounts to prevent unauthorized access.
- Clear Data Regularly: Delete old accounts, cookies, and unused apps.
- Use VPNs or Incognito Mode: Reduce passive data collection.
- Monitor Your Footprint: Google yourself or use tools like Have I Been Pawned to check for exposed data.

🌐 DIGITAL SOCIETY AND NETIZEN: NET ETIQUETTES

✅ Who is a Netizen?

A **netizen** (internet + citizen) is someone who actively participates in the online world. Netizens are responsible for contributing positively to the digital community, just like good citizens in real life.

💻 What is Net Etiquette (Netiquette)?

Netiquette refers to the code of proper behavior and manners while interacting online. It's about being **respectful, kind, and responsible** on the internet—just like good manners in the real world.

📌 Key Rules of Net Etiquette

Area	Best Practices
General Communication	Be polite, avoid typing in ALL CAPS (it's like shouting), and use proper grammar.
Respect Others	Avoid offensive comments, personal attacks, or hate speech.
Privacy	Don't share someone else's personal information or photos without consent.
Avoid Spamming	Don't flood chats, emails, or comment sections with repeated or irrelevant messages.
Constructive Feedback	Give respectful and useful feedback, especially in forums or class groups.

Reply Promptly	Respond to messages or emails within a reasonable time, especially in professional settings.
Avoid Plagiarism	Give credit to sources—don't copy-paste without acknowledgment.
Be Helpful	Assist others when they need help or information online.
Think Before You Post	Ask yourself: Is it true? Is it kind? Is it necessary?
Cultural Sensitivity	Be aware that people from different cultures might interpret your messages differently.

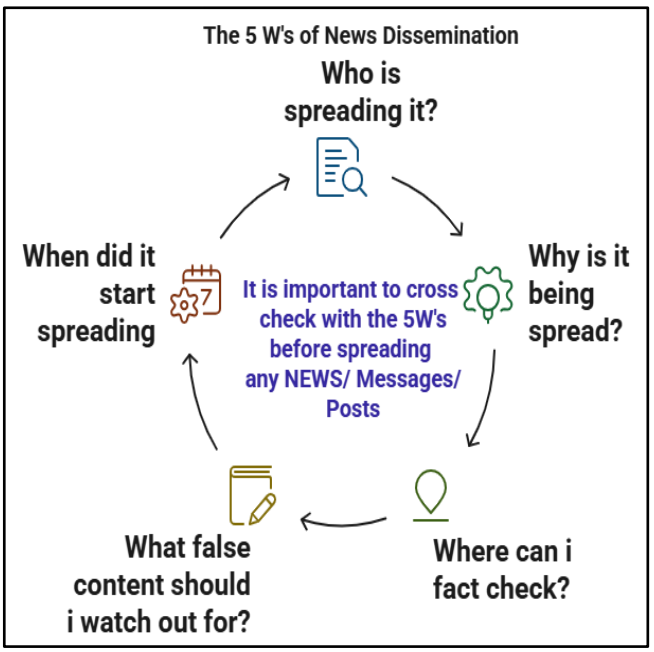
COMMUNICATION ETIQUETTES

These are the rules and guidelines for effective and respectful communication, especially in digital environments like emails, chats, and online meetings.

1. **Be Clear and Concise**: Use simple and direct language. Avoid long, confusing sentences.
2. **Use Appropriate Language:** Avoid slang, offensive words, or all-caps (which can be interpreted as shouting).
3. **Be Respectful:** Address people politely. Use greetings like *Dear Sir/Ma'am*, and closings like *Regards*.
4. **Use Subject Lines (in emails)**: Always include a meaningful subject line.
5. **Proofread**: Check spelling and grammar before sending any message.
6. **Respond Promptly**: Reply to emails and messages in a timely manner.
7. **Avoid Spamming**: Don't send unnecessary or repetitive messages.
8. Listen Actively (in online meetings): Don't interrupt, and use mute when not speaking.

SOCIAL MEDIA ETIQUETTES

These are guidelines to behave responsibly and respectfully on platforms like WhatsApp, Instagram, Facebook, Twitter, etc.



1. **Think Before Posting**: Avoid posting anything offensive, controversial, or harmful.
2. **Respect Privacy**: Don't share others' personal information or photos without permission.
3. **Avoid Cyberbullying**: Do not troll, harass, or spread rumors about anyone.
4. **Be Authentic:** Don't pretend to be someone else or spread fake news.
5. **Use Proper Language:** Maintain a decent tone, especially in public posts.
6. **Give Credit:** Acknowledge sources when sharing someone else's work or content.
7. **Manage Time:** Don't overuse social media, and take breaks when needed.
8. **Report Inappropriate Content:** Use reporting tools on platforms to flag abusive or harmful content.

DATA PROTECTION

Data protection refers to the process of safeguarding important information from corruption, loss, or unauthorized access. Examples of sensitive data include biometric information, health information, financial information, or other personal documents, images or audios or videos. The motive behind implementation of safeguard policies is to ensure that sensitive information is appropriately protected from modification or disclosure.

INTELLECTUAL PROPERTY RIGHTS (IPR)

IPR refers to the legal rights given to individuals or organizations for creations of the mind—like inventions, literary works, symbols, names, and designs. These rights help **protect ownership and encourage innovation and creativity.**

Types of Intellectual Property Rights

Type	Description	Example
Copyright	Protects **original literary, artistic, musical, or dramatic works**.	Books, songs, movies, software code
Patent	Granted for a **new invention**, giving the creator exclusive rights for a limited time.	A new medicine or machine
Trademark	Protects **symbols, logos, brand names, or slogans** that distinguish products or services.	Nike logo, "Just Do It" slogan
Design Rights	Protect the **appearance, shape, or configuration** of a product.	Design of a smartphone or a piece of furniture
Geographical Indication (GI)	refers to products that have a **specific geographical origin** and possess qualities or a reputation due to that location.	Darjeeling tea, Banarasi sarees

Violation of IPR

Type	Meaning	Example
Plagiarism	Plagiarism: Copying someone else's work or ideas and presenting them as your own without credit.	Submitting someone else's essay as your own.
Copyright Infringement	Using copyrighted material without permission.	Uploading pirated movies or songs.
Trademark Infringement	Using a registered trademark without authorization.	Selling fake Adidas shoes with the logo.

Why IPR is Important

- Encourages **innovation and creativity**
- Helps **creators earn recognition and financial benefit**
- Prevents **unauthorized use or misuse** of creative works
- Supports **economic development** by protecting business assets

Open Source and Licensing

Open Source Software is any computer software that is distributed with its source code available for modification.

Examples of OSS are Linux Unix, MySQL etc.

Criteria for the Distribution of OSS

- Free Redistribution

- Source code
- Integrity of The Author's Source Code

Open-source software is software where the **source code is freely available** for use, modification, and distribution.

License Type	Description
Creative Commons (CC)	Allows creators to specify what others can do with their work (like share or remix).
GNU General Public License (GPL)	Ensures that software is free to use, modify, and redistribute, and any modified version must also be open.
Apache License	Permissive license allowing modifications and use, even in proprietary software, with credit to the original developers.

Software License

A software license is a license agreement that gives an individual, a company or an organisation permission to use a software program.

Types of software License

(i) Proprietary Software License

It is a legal agreement that gives the user permission to use the software under specific conditions but **does not allow access to the source code** or permission to modify, share, or redistribute the software. Example: **Microsoft Windows** and **Adobe Photoshop** are proprietary software.

(ii) Free and Open-Source Software License

It refers to the software that users can safely run, adopt and redistribute without legal restraint. OSS refers to freedom to use, share and/or modify the source code. Examples: **Linux**, **LibreOffice**, **Mozilla Firefox**, **GIMP**.

Open source softwares are further classified into Permissive license and Copyleft license.

- **Permissive License**
  - MIT License
  - BSD License
  - Modified BSD License or 3-clause License
  - Simplified BSD License or 2 clause License
  - Apache License
- **Copyleft License**
  - GNU GPL (General Public License)
  - CC (Creative Common) License
  - GNU Lesser General Public License (LGPL)

CYBER CRIME

Cyber crime refers to illegal activities carried out using computers, the internet, or other digital devices. Cyber crimes include various criminal activities carried out to steal data or to break down important services. These include hacking, spreading viruses or malware, sending phishing or fraudulent emails, ransomware, etc.

💻 Hacking

Hacking is **unauthorized access** to a computer, network, or digital system.

- Often done to **steal**, **alter**, or **destroy** data.
- Involves **breaking into systems without permission**.
- Hackers may:
  - Steal personal information
  - Damage or delete files
  - Disrupt online services
- **Prevention tips**:
  - Use **strong passwords**

- Install **firewalls**
- Keep **antivirus software** updated

👂Eavesdropping

Eavesdropping means **secretly intercepting** digital communication.

- Attackers listen to **emails, chats, or data transfers** without permission.
- Common on **unsecured networks** (e.g., public Wi-Fi).
- Goal: Steal **sensitive** info like:
  - Passwords
  - Credit card numbers
  - Private messages
- **Prevention tips:**
  - Use encryption
  - Access only HTTPS websites
  - Connect via a VPN

Phishing and Fraud Emails

- **Phishing**: Fake emails that trick users into revealing personal info (e.g., passwords, bank details).
- **Fraud emails**: Scam messages offering fake prizes, jobs, or money to deceive people.
- Often include **fake links**, **urgent messages**, or **too-good-to-be-true offers**.
- **Goals**:
  - Steal data
  - Spread malware
  - Trick users into sending money

Safety Tips:

- Don't click suspicious links.
- Never share personal or financial info.
- Use spam filters and report phishing attempts.

HOW TO HANDLE SUSPICIOUS EMAIL

1. If you receive a phishing email message, do not respond to it. Don't open junk mail.
2. Approach links in e-mail messages with caution.
3. Approach images in email with caution Approach attachments in e-mail messages with caution.
4. Don't trust the sender information in an e-mail message. Don't trust offers that seem too good to be true. Report suspicious email.
5. Don't enter personal or financial information into pop-up windows. Don't forward chain email messages.
6. Update your computer software.
7. Verify the Identity and Security of Websites.
8. Type addresses directly into your browser or use your personal bookmarks.

🛑Ransomware

A type of malware that locks or encrypts your files.

- The attacker demands a ransom (money) to restore access.
- Spread through malicious emails, links, or downloads.
- Victims can lose access to important data.

- **Prevention Tips**:
  - Keep backups of important files.
  - Use updated antivirus software.
  - Avoid clicking unknown links or attachments.

🔒 **Eg:** In the **WannaCry** attack (2017), ransomware spread worldwide, encrypting data in hospitals, businesses, and governments, demanding Bitcoin as ransom.

😈 Cyber Troll

Cyber trolling is the act of **posting offensive, hurtful, or provocative messages online** to upset others.

- Trolls often target people on **social media, forums, or comment sections**.
- The goal is to **provoke**, **insult**, or cause **emotional harm**.
- It can lead to **cyberbullying** or mental stress for victims.

Safety Tips:

- **Ignore** and **do not respond** to trolls.
- **Report** and **block** abusive users.
- Use **privacy settings** to control who interacts with you online.

🚫 Cyberbullying

- **Cyberbullying** is when someone uses **digital platforms** (like social media, messages, or emails) to **harass, threaten, or humiliate** others.
- It can include **spreading rumors**, **sending hurtful messages**, or **posting embarrassing content**.
- Victims may feel **depressed, anxious, or scared**.

Prevention & Safety Tips:

- **Do not respond** to bullies.
- **Block and report** the bully.
- **Talk to a trusted adult** or teacher.
- Keep **evidence (screenshots)** of the abuse if needed.

🛡️ CYBER SAFETY

Cyber safety refers to practices and precautions that protect users from online threats and ensure a safe digital experience.

🌐 Safely Browsing the Web

1. **Use Secure Websites:**
   Look for https:// and a 🔒 lock symbol in the address bar.
2. **Avoid Suspicious Links:**
   Don't click on unknown ads, pop-ups, or download attachments from untrusted sources.
3. **Install Antivirus/Antimalware Software:**
   Helps detect and remove malicious software.
4. **Keep Software Updated:**
   Regular updates fix security holes in browsers and apps.
5. **Use Search Engines Wisely:**
   Don't enter personal info on random websites.
6. **Avoid Using Public Wi-Fi for Sensitive Tasks:**
   Hackers can intercept your data on open networks.

👦💼 Identity Protection

1. **Keep Personal Information Private:**
   - Do not share your full name, address, passwords, or school name publicly.
2. **Use Strong Passwords:**
   - Use a mix of letters, numbers, and symbols.
   - Avoid using names, birthdates, or "123456".
3. **Enable Two-Factor Authentication (2FA):**
   - Adds an extra layer of protection when logging in.
4. **Beware of Phishing Scams:**
   - Emails or messages pretending to be from trusted sources asking for personal info.

🔐 Confidentiality

1. **Encrypt Sensitive Data:**
    - Use encryption tools or secure platforms when sharing sensitive info.
2. **Don't Share Passwords:**
    - Even with friends. Keep login details confidential.
3. **Log Out After Use:**
    - Especially on shared or public devices.
4. **Access Control:**
    - Only authorized users should be allowed to access important data.
5. **Respect Others' Privacy:**
    - Never snoop into others' accounts or files.

🦠 MALWARE

- **Malware** (malicious software) is designed to **harm or gain unauthorized access** to computer systems.
- Common types:
    - **Viruses** – replicate and spread by attaching to files.
    - **Trojans** – disguised as legitimate software.
    - **Adware** – displays unwanted ads and may track user activity.
- Knowing the types helps in **prevention and protection**.

VIRUSES

- A **virus** is a type of **malware** that attaches to **legitimate programs or files**.
- It **replicates** when the infected file or program is run.
- Spreads through **email attachments**, **infected files**, or **USB drives**.
- Can **corrupt/delete data**, **slow down systems**, and cause other damage.

🐴 TROJANS (Trojan Horses)

- **Trojans** disguise as **legitimate software** to trick users into installing them.
- Unlike viruses, **they do not replicate** themselves.
- Use **social engineering** (fake updates, apps, etc.) to spread.
- Once installed, they can:
    - **Steal sensitive data**
    - **Create backdoors** for hackers
    - **Install more malware**

ADWARE

- **Adware** shows or downloads **ads** automatically while you're online.
- Not always harmful, but can:
    - **Slow down** your system
    - **Clutter** the screen with unwanted ads
- Some adware may **track user activity** and **collect personal data**, causing **privacy concerns**.

E-WASTE MANAGEMENT

♻️ E-Waste (Electronic Waste)

E-waste refers to discarded electrical or electronic devices and components that are no longer useful or working.

E-waste broadly covers waste from all electronic and electrical appliances and comprises items such as computers, mobile phones, digital music recorders/ players, refrigerators, washing machines, televisions (TVs) and many other household consumer items.

A device is declared as **e-waste (electronic waste)** when it

- **It is no longer working** and cannot be repaired. *Example:* A smartphone with a damaged motherboard that can't be fixed.
- **It is outdated or obsolete**, even if still functional. *Example:* CRT monitors, floppy disk drives, or early-generation printers.
- **It has been replaced by a newer model**, and the old device is no longer in use. *Example:* Upgrading to a smart TV and discarding the old LCD TV.
- **It poses environmental or health risks** due to the hazardous materials inside. *Example:* Batteries leaking harmful chemicals, or devices with exposed wiring.
- **It is broken or damaged beyond use**, and the cost of repair is higher than replacement. *Example:* A laptop with severe water damage.

E-WASTE HAZARDS

- Most electronic waste contains toxic chemicals, such as lead, beryllium, and mercury.
- Improper disposing of gadgets and devices causing air and water pollution.
- Improper e-waste recycling is causing health issues like: Damage to the immune system, Skin disease, Skin problems.

E-waste management requires proper recycling and recovery of the disposed material. The recycling and recovery process includes the following steps.

1. **Dismantling**: - removal of parts containing valuable items such as- copper, silver, gold, steel and removal of parts containing dangerous substances like- mercury, lead, Beryllium etc.
2. **Separation of metal and plastic**
3. **Refurbishment and reuse:** - it means used electrical and electronic items that can be easily remodel to make it's to reuse
4. **Recovery of valuable materials**
5. **Disposal of dangerous materials** like- mercury, lead, Beryllium etc and disposed off in underground landfill sites.

✓ E-Waste Management Tips

Practice	Description
Reduce	Avoid unnecessary upgrades; use devices longer.
Reuse	Donate or sell old but working electronics.
Recycle	Give e-waste to certified recycling centers.
Repair	Fix broken electronics instead of discarding them.
Buy Green	Choose energy-efficient and eco-friendly products.

INFORMATION TECHNOLOGY ACT (IT ACT) 2000

It is **India's first law** to give legal recognition to **electronic transactions**.
Addresses **cybercrimes**, **data protection**, and **electronic governance**.
Amended in **2008** to include provisions for:

- **Data privacy**
- **Sensitive information processing, storage, and transmission**

Provides a **legal framework** for:

- **Electronic records**
- **Digital signatures**
- **Online contracts and governance**

Defines various **cybercrimes** and their **penalties** (e.g., hacking, identity theft, data breach).
Cyber cells have been set up in many states to help people report cybercrimes.

Objectives of the IT Act:

1. To legalize digital signatures and electronic records
2. To prevent and punish cyber crimes

3. To promote safe and secure use of computers and the internet
4. To facilitate e-governance and online business transactions

TECHNOLOGY AND SOCIETY: GENDER AND DISABILITY ISSUES WHILE TEACHING AND USING COMPUTERS

ICT (Information and Communication Technology) are general purpose technologies whose values and impact arise primarily from their use in other economic and social sectors.

Gender Issues in Teaching and Using Computers:

- **Stereotypes and Bias**: Girls are sometimes seen as less capable in tech fields, which can discourage them from participating actively in computer education.
- **Access and Encouragement**: In some places, boys may be given more opportunities, resources, and encouragement to use computers compared to girls.
- **Curriculum Content**: Learning material may unintentionally reflect male-dominated examples or not highlight female role models in tech.
- **Confidence Gap**: Girls might feel less confident using computers due to social pressure, even when they have the same or better abilities.
- **Participation Rates**: Fewer girls might choose computing-related subjects or careers, leading to gender imbalance in the tech industry.

Disability Issues in Teaching and Using Computers:

- **Accessibility Challenges**: Many computer systems, websites, and software are not fully accessible to students with disabilities (like visual, hearing, or motor impairments).
- **Assistive Technologies**: There is often a lack of proper assistive tools (like screen readers, alternative keyboards, speech recognition software) in classrooms.
- **Teacher Training**: Teachers may not be trained adequately to adapt lessons for students with disabilities.
- **Physical Setup**: Computer labs might not be designed to accommodate students who use wheelchairs or other mobility aids.
- **Learning Materials**: Content may not be provided in formats that suit all disabilities (e.g., audio materials for visually impaired learners).

Teaching Considerations:

- **Inclusive Language:**
  Use gender-neutral language and avoid perpetuating stereotypes when teaching computer concepts.
- **Accessibility Features:**
  Ensure all learning materials and resources are accessible to individuals with disabilities, including providing alternative formats, captions, and screen reader compatibility.
- **Assistive Technologies:**
  Be familiar with and able to utilize assistive technologies to support students with disabilities.
- **Adaptations and Modifications:**
  Be prepared to adapt teaching methods and assignments to meet the individual needs of students with disabilities.
- **Promoting Diversity and Inclusivity:**
  Create a learning environment that values diversity and inclusivity, where all students feel welcome and supported

QUESTION BANK

MULTIPLE CHOICE QUESTIONS

1. **Which of the following is considered a violation of intellectual property rights?**
   a) Copy-pasting content from a website without credit
   b) Sharing a meme online
   c) Using open-source software with permission
   d) Creating original content
2. **What is the primary goal of net etiquettes?**
   a) Promoting malware awareness
   b) Ensuring polite and respectful online communication
   c) Protecting against phishing
   d) Increasing social media followers
3. **Which license allows users to freely use, modify, and distribute software with few restrictions?**
   a) Trademark License
   b) Creative Commons (CC)
   c) GNU General Public License (GPL)
   d) Proprietary License
4. **Which of the following is NOT considered a form of cybercrime?**
   a) Phishing
   b) Cyberbullying
   c) Ransomware
   d) Creative Commons Licensing
5. **Which act in India provides a legal framework for electronic governance and cybercrime?**
   a) Data Protection Act
   b) Indian Penal Code
   c) Information Technology Act
   d) Cyber Safety Act
6. **What is malware?**
   a) S/W used to manage computer files
   b) S/W intended to damage/disable PC
   c) A legal software license
   d) A communication protocol
7. **E-waste refers to:**
   a) Old books and magazines
   b) Outdated electronics that are discarded
   c) Recyclable plastic only
   d) Broken furniture
8. **Which of the following is considered safe web browsing practice?**
   a) Using antivirus software
   b) Sharing passwords via email
   c) Clicking unknown pop-up ads
   d) Downloading s/w from unverified sources
9. **Which of these is an example of identity theft?**
   a) Logging into your own account
   b) Using someone else's photo as your profile picture
   c) Impersonating someone online using their personal details
   d) Posting a meme
10. **Which software license is most often applied to academic and creative content?**
    a) Apache
    b) Creative Commons
    c) Proprietary
    d) Microsoft EULA
11. **What is an example of proper net etiquette when communicating online?**
    a) Using all caps to emphasize a point
    b) Sending chain emails to multiple recipients
    c) Posting personal information of others without consent
    d) Responding politely and respectfully in discussion
12. **Which of the following is a violation of Intellectual Property Rights (IPR)?**
    a) Using open-source software under Creative Commons license
    b) Copying and distributing a copyrighted movie without permission
    c) Creating original content and sharing it online
    d) Using a trademark with the owner's consent
13. **Shyam told his friend Priya about a term that "It is the study of ethics pertaining to computer, encompassing user behavior and what computers**

are programmed to and how this affects individuals and society" What is it?

a) Cyber ethics b) Plagiarism c) Netiquette d) Hacking

14. **Which of the following refers to an attempt to gain information from protected or unauthorized parts of a computer system or network.?**
    a) Hacking b) Knowledge c) Etiquette d) Plagiarism
15. **What is phishing in the context of cybercrime?**
    a) Encrypting files to demand a ransom
    b) Pretending to be a trustworthy entity to steal sensitive information
    c) Overloading a website with traffic to disrupt it
    d) Spreading viruses through infected software
16. **Which practice contributes to effective e-waste management?**
    a) Disposing of old electronics in regular trash bins
    b) Recycling electronic devices at certified facilities
    c) Storing unused gadgets indefinitely in homes
    d) Burning old electronic components
17. **Under the Information Technology Act (IT Act), which activity is considered illegal?**
    a) Sharing open-source software with proper licensing
    b) Creating a website for educational purposes
    c) Using social media to post personal opinions
    d) Unauthorized access to someone's computer system

ANSWERS

1	2	3	4	5	6	7	8	9
a	b	c	d	c	b	b	a	c
10	**11**	**12**	**13**	**14**	**15**	**16**	**17**	
b	d	b	a	a	b	b	d	

ASSERTION AND REASONING QUESTIONS

In the following question, 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 the correct explanation of A.
B. Both A and R are true, but R is NOT the correct explanation of A.
C. A is true, but R is false.
D. A is false, but R is true.

1. **Assertion (A):** Ransomware encrypts user data and demands payment for decryption.
   Reason (R): Ransomware is a type of open-source software.
2. **Assertion (A):** The IT Act protects users against cyber fraud and crime in India.
   Reason (R): It was enacted to regulate online transactions and ensure cybersecurity.
3. **Assertion (A):** Social media etiquettes help maintain a positive online environment.
   Reason (R): Trolling and abusive behavior are not encouraged on social platforms.
4. **Assertion (A):** Using open-source software increases the risk of copyright infringement.
   Reason (R): Open-source licenses clearly define usage and distribution rights.
5. **Assertion (A)**: Using someone else's trademark without permission can lead to legal consequences.
   Reason (R): Trademarks are protected under Intellectual Property Rights to prevent unauthorized use.

6. **Assertion (A)**: Cyberbullying is a form of cybercrime that involves harassing someone online.
   Reason (R): Cyberbullying can cause emotional harm but is not considered a criminal offense.
7. **Assertion (A)**: Gender and disability issues are irrelevant when teaching computer skills.
   Reason (R): Technology education must be inclusive to ensure equal access and opportunities for all.
8. **Assertion (A):** Recycling e-waste can help conserve natural resources.
   Reason (R): Most e-waste is generated in rural areas where technology use is minimal.
9. **Assertion (A):** Digital footprints can be completely erased by deleting browsing history.
   Reason (R): Websites and online services may store user data on their servers even after it is deleted locally.
10. **Assertion (A):** Intellectual Property Rights (IPR) encourage innovation and creativity.
    Reason (R): Trademark protects logos and brand names from unauthorized use.

ANSWERS

1	2	3	4	5	6	7	8	9	10
C	A	B	D	A	C	D	B	D	B

SHORT ANSWER QUESTIONS

Q1. Differentiate between virus and trojan.
Q2. Discuss any five ways to manage and reduce e-waste.
Q3. Compare copyright, patent, and trademark.
Q4. Distinguish between active digital footprint and passive digital footprint.
Q5. Explain Intellectual Property Rights. How is plagiarism a violation of IPR?
Q6. Mention any two effects of cyberbullying.
Q7. How can spyware harm you?

Answer1: Virus: A self-replicating program that infects files and spreads.
Trojan: Malicious software that appears to be legitimate but creates a backdoor and steals data once installed.

Answer2: a) Recycle old devices properly. b) Donate usable gadgets.
c) Avoid unnecessary upgrades. d) Use certified e-waste recycling centers.
e) Buy environmentally-friendly electronics**.**

Answer3:

Aspect	**Copyright**	**Patent**	**Trademark**
Protects	Original literary/artistic work ,software	Inventions and innovations	Logos, brand names, slogans
Validity	Life of author + 60 years	20 years from filing date	Can be renewed indefinitely
Example	Book, music, software code	New machine design	Nike logo, Apple name

Answer4.

Active Digital footprint	Passive digital footprint
* An active digital footprint is where the user has deliberately shared information about themselves either by using social media sites or by using websites *The most obvious example is sharing information on social media , but email also contributes to your active footprints	*A passive digital footprint is a data trail you unintentionally leave online. *For example your IP address approximate location or browser history.

Answer 5:
IPR refers to the legal rights granted to creators for their inventions, literary works, symbols, and designs.
Plagiarism violates copyright laws by copying someone else's content without credit, thus infringing on their IPR.

Answer 6: Cyberbullying can harm a person's emotional and mental health.
It may lead to anxiety, depression, and social withdrawal.
Victims often feel unsafe and targeted even in their homes.
Awareness, support, and strict laws can help prevent it.

Answer7:-Spyware can harm you in many ways such as : -

- Malware will log your keystrokes.
- Steal your passwords.
- Observe your browsing choices.
- Send your targeted E mail.
- Redirect your web browser to phishing pages.
- Can affect the performance of your computer system.
- Report your personal information to phishing pages.

CASE STUDY BASED QUESTIONS

Q1. Anshika is a computer science student who has developed a photo editing app as part of her school project. She decided to upload her app to a public repository for others to use and improve. While exploring similar apps, she finds another app that looks almost identical to hers, including the user interface and some exact code blocks she wrote. The other developer did not give her credit or ask for permission. Meanwhile, Anshika had licensed her app under the **GPL (General Public License)**. She also notices that the other developer has used her app's name and logo. She is now considering the legal and ethical aspects of the issue.

(a) Identify **two types of Intellectual Property Rights (IPRs)** that have possibly been violated in Anshika's case.
(b) What type of license is the GPL, and what does it allow or restrict?
(c) If Anshika had used a **Creative Commons License**, what flexibility might she have had in terms of allowing modifications or sharing?
(d) Suggest two actions Anshika can take to protect her work from future violations.
(e) How this case is of copyright infringement and trademark infringement? Justify your answer.

Q2. Priya is a high school student who enjoys sharing her travel photos and daily experiences on social media platforms. Recently, she received a friend request from a

stranger who claimed to be a fellow travel enthusiast. Excited, she accepted the request and shared more personal details like her location and school name. A week later, her parents noticed strange messages on her phone and found that someone was impersonating her online using her photos and personal information. Based on the above case, answer the following:

(a) Identify and explain the type of cybercrime Priya has experienced.

(b) What mistakes did Priya make that led to this situation?

(c) Suggest any two safety measures Priya can take to protect her digital footprint in the future.

Q3. Meena is a budding artist who regularly uploads her original digital artwork to social media platforms to showcase her talent. Recently, she discovered that some of her artwork was copied by an unknown seller, printed on t-shirts and mugs, and sold online without her permission or giving her any credit. Meena felt cheated and began researching her rights as a creator and how to legally protect her work.

Answer the following:

(a) What type of cybercrime is depicted in this situation?

(b) Why is it important for creators like Meena to protect their digital content?

(c) Suggest **two preventive measures** Meena can take to avoid misuse of her artwork in the future.

(d) Name any **two Indian laws** or acts that can help Meena in such cases.

(e) If Meena wants to take legal action, what should be her **first step**?

ANSWERS

Answer 1:

(a) Two types of **Intellectual Property Rights (IPRs)** that may have been violated:

1. **Copyright** – because the other developer copied Anshika's code and user interface without permission.
2. **Trademark** – because the developer used the same app name and logo.

(b) The **GPL (General Public License)** is a type of **open-source license** that:

- Allows users to use, modify, and distribute the software.
- Requires that any modified version also be distributed under the **same GPL license**.
- Ensures the software and its derivatives remain open-source.
- Credit to the original author must be given.

(c) If Anshika had used a **Creative Commons License**, she could:

- Choose whether to allow or restrict commercial use.
- Decide whether to permit derivative works.
- Require attribution, share-alike conditions, or non-commercial usage.
- Have more flexibility in defining the terms of sharing and modification.

(d) Two actions Anshika can take to protect her work:

Register her copyright and trademark officially to enforce legal rights.

Use proper open-source licensing with clear terms and include a license file in her code repository.

(e) This case involves copyright infringement for using her code and design without permission, and trademark infringement for using the same app name and logo.

Answer 2:

(a) The cybercrime Priya experienced is **Identity Theft** and **Cyber Impersonation**. This happens when someone uses another person's personal details, like photos or identity, to create fake profiles or commit fraud online.

(b) Mistakes Priya made:

- She **accepted a friend request from a stranger** without verifying their identity.
- She **shared sensitive personal details**, like her **location and school**, on a public platform.

(c) Two safety measures Priya can take:

1. **Set social media profiles to private** and only accept connection requests from known people.
2. **Avoid sharing sensitive information** like location, school name, or personal photos publicly on the internet.

Answer 3: -

(a) The type of cybercrime is **Copyright Infringement** or **Intellectual Property Theft**.

(b) It is important because:

- It helps protect ownership rights.
- Prevents unauthorized commercial use of their work.
- Ensures creators receive due credit and possible earnings.

(c) Two preventive measures:

1. **Add watermarks** or signatures to all uploaded digital artwork.
2. **Register the artwork under copyright** to establish legal ownership.

(d) Two Indian laws that can help:

1. **The Copyright Act, 1957**
2. **The Information Technology Act, 2000**

(e) First step for legal action:

1. Meena should **file a complaint with the Cyber Crime Cell** or contact a lawyer specializing in intellectual property law.

SAMPLE QUESTION PAPER WITH SOLUTION (SET 1)

Class: XI	**Max Marks: 70**
Sub: Computer Science (083)	**Time Allowed: 3 Hrs.**

General Instructions:

1. The question paper contains 37 questions.
2. All questions are compulsory.
3. The paper is divided into 5 Sections- A, B, C, D and E.
4. Section A consists of 21 questions (1 to 21). Each question carries 1 Mark.
5. Section B consists of 7 questions (22 to 28). Each question carries 2 Marks.
6. Section C consists of 3 questions (29 to 31). Each question carries 3 Marks.
7. Section D consists of 4 questions (32 to 35). Each question carries 4 Marks.
8. Section E consists of 2 questions (36 to 37). Each question carries 5 Marks.
9. All programming questions are to be answered using Python Language only.
10. In case of MCQ, text of the correct answer should also be written.

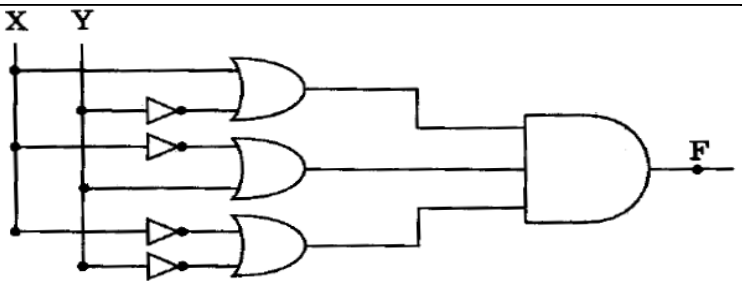
SECTION – A		
Q. No.	**Question**	**Marks**
1.	What is the primary function of the Control Unit (CU) in the CPU? (a) Managing Data Storage (b) Performing Calculations (c) Coordinating and controlling other components in the CPU (d) Accessing data from the secondary storage	1
2.	How many MB is there in 2TB? (a) 2^{21} (b) 2^{20} (c) 2^{31} (d) None of the above	1
3.	The base of binary number system is …… (a) 2 (b) 6 (c) 7 (d) 8	1
4.	State TRUE or FALSE Idempotent law states that X.X'=X and X+X'=X	1
5.	What shape represents a process in a flowchart? (a) Parallelogram (b) Oval (c) Rhombus (d) Rectangle	1
6.	What do we use to define a block of code in the Python language? (a) Key (b) Brackets (c) Indentation (d) None of these	1

7.	What will be the output of the following code?
	x = 500//2
	if x < 250:
	  print("Less than 250")
	elif x > 250:
	  print("Greater than 250")
	else:
	  print("Greater than or equal to 250")
	(a) Less than 250 (b) Greater than 250
	(c) Greater than or equal to 250 (d) Error
		1

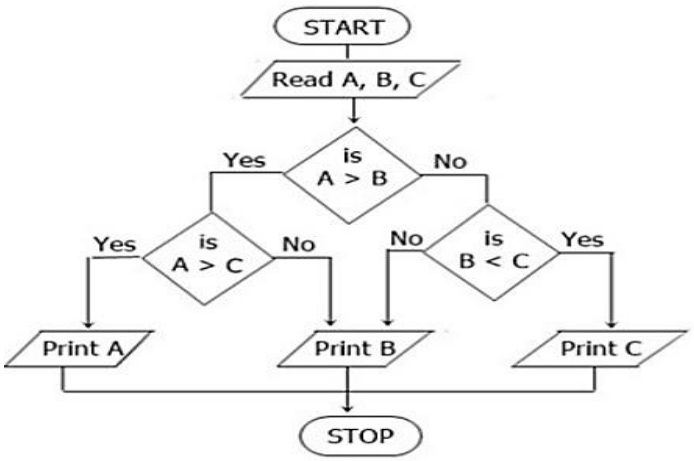
8.	Which of the following is not a valid keyword of Python associated with loops? (a) continue (b) check (c) range (d) break	1
9.	What is the output of print("hello"+"1+2+3") (a) hello1+2+3 (b) hello6 (c) hello123 (d) Error	1
10.	Which of the following function returns lowest item in a list LST?	1

	(a) sum(LST) (b) len(LST) (c) cmp(LST) (d) min(LST)	
11.	A variable items in a Python program is defined as: T = 1, 2, 3, 4 What type of data type is used for the T variable? (a) Tuple (b) List (c) Dictionary (d) String	1
12.	What will be the output of following Python code? tp1=(5,10) print(tp1*2) (a) (5,10) (b) (10,20) (c) (5,10,5,10) (d) Error	1
13.	What will be the output of the following code: d={1:"Green",2:"Blue",3:"Orange"} print("Green" in d) (a) True (b) False (c) Error (d) None	1
14.	A Python module has ---------- extension. (a) .mod (b) .imp (c) .py (d) .mpy	1
15.	Which module is used to use mean() function? (a) random() (b) randint() (c) uniform() (d) statistics	1
16.	When someone steals someone else's personal information to commit theft or fraud, it is called…….. (a) Hacking (b) Infringement (c) Identity theft (d) None	1
17.	Exploring appropriate and ethical behaviours related to online environments and digital media (a) Cyber Security (b) Cyber Safety (c) Cyber Law (d) Cyber Ethics	1
18.	Which of the following is the type of software that has self-replicating software that causes damage to file and system? (a) Trojans (b) Viruses (c) Worm (d) Adware	1
19.	Which of the following is not a proper e-waste disposal technique? (a) Dismantling (b) Segregation (c) Recycling (d) Throwing away	
Q.20 and 21 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 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		
20.	**Assertion (A):** Indentation refers to adding the relevant number of tabs and spaces at the beginning of lines of code to indicate a block of code. **Reason (R):** Indentation is very important to make the code readable, without indentation you find it hard to read or debug a program.	1
21.	**Assertion (A):** List is an immutable data type. **Reason (R):** When an attempt is made to update the value of an immutable variable, the old value is destroyed and a new variable is created by the same name in memory.	1
SECTION – B		
22.	Convert the following (i) $(101)_{10}$ = (……..)$_2$ (ii) $(A1B.25)_{16}$=(……..)$_8$	1 1
23.	Draw a flowchart to find the biggest among three numbers. **OR** Design an algorithm to find the factorial of a number.	2
24.	Evaluate the following: x,y,z=3,12,5 y,z,x=x+2,y-2,z+5	2

	print(x,y,z)	
25.	Identify and correct syntax errors (if any).Underline each correction. 30=To for K in range(0,To) IF K%4==0: print (K*4) Else: print (K+3) **OR** Give the output of the following Python expressions. (i) print(10!=9 and 20>=20) (ii) print(10+6*2**2!=9//4-3 and 29>=29/9)	2
26.	Consider the following string given below and answer the following questions: s=”programming is an art” (a) print(s[3:7]*3) (b) print(s.partition("is")	2
27.	Explain the following terms: (a) Trojan Horse (b) IPR	2
28.	Write at least two Gender and two Disability issues faced while teaching and using computers. **OR** What do you understand by the term Intellectual property rights?	2
	SECTION – C	
29.	Write a Python program using nested loop to produce the following pattern: A A B A B C A B C D A B C D E	3
30.	Meera has written the following code to create a tuple with a single item. T=(10) X=len(T) But when she tried to get the length of the tuple, Python gives the following error. TypeError: object of type ‘int’ has no len() (a) What may be the reason? Help her to solve the problem (b) What will be the output of the following print statement? T1=(10,20,30) T2=(15,25,35) print(T1+T2) (c) Write the output of the following code snippet. T3=(1,2,3) for i in T3: print(i*5)	3
31.	Why is it important to recycle e-waste? Give any 3 distinct points.	3
	SECTION – D	
32.	(i) State DeMorgan’s Theorems. (ii) Write the equivalent Boolean expression for the following Logic Circuit:	2+2 =4

		
33.	(a) Explain the following string functions with example. (i)title() (ii) find() (b) Write a Python program to count the number of vowels in a string. **OR** Write a Python program to input a string and (i) To print the number of upper and lower case letters in it (ii) To check if it contains digit or not.	2+2 = 4
34.	Write a Python program that repeatedly ask the user to enter the book names and prices. Store all of them into a dictionary whose keys are book name and values are prices. After entering all books and prices display the dictionary and those books whose price is more than Rs.750.	4
35.	Explain different Cyber-crimes for which cyber laws are enforced in strictly in India.	4
	SECTION – E	
36.	The record of a student (Name, Roll NO., Marks in five subjects and percentage of marks) is stored in the following list. sturec=["Devan",14,[56,98,99,72,69],79.5] Write Python statements to retrieve the following information from the list sturec. i. Percentage of the student ii. Marks in fourth subject iii. Maximum marks of the student iv. Remove the percentage from sturec v. Change the name of the student from "Devan" to "Jai Dev"	5
37.	(i) Go through the python code shown below and find out the possible output(s) from the suggested options (a) to (d). What is the maximum and minimum values of the variable R import random signal=["RED","YELLOW","GREEN"] for K in range(2,0,-1): R=random.randrange(K) print(signal[R],end="#") (a)YELLOW#RED# (b) RED#GREEN# (c) GREEN#RED# (d) YELLOW#GREEN# (ii)What is a Python module? What is the significance? **OR** Write a program in Python to input five state names and their capital in a dictionary as Key Value pair, then ask the user to enter state and print its capital if found in a dictionary, otherwise print message "State not found in dictionary"	3+2 =5

ANSWERS (SET 1)

SECTION – A		
Q. No.	**Answers**	**Marks**
1.	(c) Coordinating and controlling other components in the CPU	1
2.	(a) 2^{21}MB	1
3.	(a) 2	1
4.	False	1
5.	(d) Rectangle	1
6.	(c) Indentation	1
7.	(c) Greater than or equal to 250	1
8.	(b) check	1
9.	(a) hello1+2+3	1
10.	(d) min(LST)	1
11.	(a) Tuple	1
12.	(c) (5,10,5,10)	1
13.	(b) False	1
14.	(c) .py	1
15.	(d) statistics	1
16.	(c) Identity theft	1
17.	(d) Cyber Ethics	1
18.	(c) Worm	1
19.	(d) Throwing away	1
20.	(a) Both (A) and (R) are true and (R) is the correct explanation for (A)	1
21.	(d) (A) is false but (R) is true	1
SECTION – B		
22.	(i) $(101)_{10} = (1100101)_2$ (ii) $(A1B.25)_{16}=(5033.112)_8$	1 1
23.	 **OR** STEP 1: START STEP 2 : READ N STEP 3: FACT=1 STEP 4: TERM=1 STEP 5: FACT=FACT*TERM STEP 6: CHECK TERM==N: IF YES STEP 8 ELSE STEP 7 STEP 7: C=C+1: REPEAT STEP 5 ONWARDS STEP 8: PRINT FACT STEP 9: STOP	2
24.	10 5 10	2

25.	To=30
	for K in range(0,To):
	  if K%4==0:
	  print (K*4)
	  else:
	  print (K+3)
	OR
	(i) True (ii) True
		2

26.	(a) gramgramgram (b) ('programming ', 'is', ' an art')	1 1
27.	(a) Trojan Horse is a program that appears harmless such as a text editor or a utility program but actually performs malicious functions such as deleting or damaging files. (b) Intellectual property rights (IPR) are the rights of the owner of information do decide how much information is to be exchanged, shared or distributed. Also it gives the owner a right to decide the prize for doing (exchanging/sharing/distributing) so.	1 1
28..	Gender Issues while teaching and using computers: 1. Under Representation (a). Preconceived Notions (b). Lack of interest (c). Lack of Motivation (d). Lack of Role Models (e). Lack of encouragement in class 2. Not Girl Friendly Work-Culture Disability Issues while teaching and using computers: 1. Unavailability of Teaching Materials /Aids 2. Lack of Special Needs Teachers 3. Lack of Supporting Curriculum **OR** Intellectual property rights (IPR) refer to the legal rights that protect creations, innovations, and inventions including copyrights, patents and trademarks. IPR is the rights of the owner of information to decide how much information is to be exchanged, shared or distributed. Also it gives the owner a right to decide the price for doing exchanging/sharing/distributing so.	2
SECTION – C		

29.	#Pattern Printing Nested Loop
	print("Pattern Printing using Nested Loop")
	for i in range(1,6):
	  for j in range(65,65+i):
	  print(chr(j),end=" ")
	  print()
		3

30.	(a) Tuple with single element is created as T=(10,) or T=10, if comma is not put it will be considered as int not as a sequence. (b) (10, 20, 30, 15, 25, 35) (c) 5 10 15	1+1+1 =3
31.	Recycling e-waste is crucial for several reasons: Write any 3 points. 1. Environmental Protection E-waste contains hazardous materials like lead, mercury, and cadmium that can contaminate soil, water, and air if not disposed of properly. Recycling helps to safely manage these toxins and prevent environmental pollution.	1+1+1 =3

	2. Conservation of Natural Resources E-waste recycling helps to recover valuable materials like copper, gold, and rare earth metals. By reusing these materials, we can reduce the need for mining and conserve natural resources. 3. Reduction of Greenhouse Gas Emissions The production of new electronics requires significant amounts of energy, which contributes to greenhouse gas emissions. Recycling e-waste can help reduce the energy needed to produce new devices, thereby decreasing emissions. 4. Protection of Human Health Improper disposal of e-waste can lead to exposure to toxic substances, posing health risks to humans, especially in developing countries where e-waste is often dumped. Recycling e-waste in an environmentally responsible manner helps protect human health and safety.	
SECTION – D		
32.	(i) DeMorgan's First Theorem: (X+Y)' = X'Y' DeMorgan's Second Theorem: (XY)' = X'+Y' (ii) X+Y'. X'Y. X'+Y'	1+1+2 = 4

33.	(a)
	(i)title():- it returns the all the words start with capital letters and all remaining characters in small letters.
	print("DONE IT").title()
	Output is
	Done It
	(ii) find():-returns the number of occurrences of the substring in a string.
	"abcabcdef".count("ab")
	Output is
	2
	(b)
	#python program to find the number of vowels in a string
	print("python program to find the number of vowels in a string")
	s=input("Enter the String:")
	c=0
	v="aeiouAEIOU"
	for ch in s:
	if ch in v:
	c+=1
	print("no.of vowels in the string is ",c)
	OR
	(i) #to input a string and print number of upper and lower case letters in it.
	print("input a string and print number of upper and lower case letters")
	ucase,lcase=0,0
	str1=input("enter a string:")
	for ch in str1:
	if ch>="A" and ch<="Z":
	ucase+=1
	if ch>="a" and ch<="z":
	lcase+=1
		2+2 = 4

	print("No.of uppercase characters=",ucase)
	print("No.of uppercase characters=",lcase)
	(ii) #to input a string and check whether it contains digit
	print("to input a string and check whether it contains digit")
	str1=input("enter a string:")
	test=False
	dig="0123456789"
	for ch in str1:
	  if ch in dig:
	  print("the string contains a digit")
	  test=True
	  break
	if test==False:
	  print("the string does not contains a digit")	

34.	'''python program to create a dictionary & print #books whose priceare above Rs.750'''
	print("python program to create a dictionary & print books whose price are above Rs.750")
	any="y"
	bd=dict()
	while any=="y" or any=="Y":
	  bname=input("Enter the Book Name:")
	  bprice=float(input("Enter the Book Price:"))
	  bd[bname]=bprice
	  print("This Book details are stored in the dictionary")
	  any=input("Do you want to add another book(y/n):")
	print("Details of Books in the Dictionary Format:")
	print(bd)
	print("The Details of the Books whose Price is more than Rs.750")
	for key in bd:
	  if bd[key]>750:
	  print("Book Name:",key)
	  print("Book Price:",bd[key])
		4

35.	In India, cyber laws are enforced strictly for various crimes, including: 1. Hacking: Unauthorized access to computer systems or networks. 2. Phishing: Attempting to obtain sensitive information, such as passwords or credit card details, by deceiving individuals. 3. Identity Theft: Stealing or misusing someone's identity online. 4. Cyber stalking: Harassing or threatening individuals online. 5. Online Defamation: Spreading false information about someone online to harm their reputation. 6. Child Pornography: Creating, distributing, or possessing child pornography online. 7. Cyberbullying: Harassing or intimidating individuals online, often through social media. 8. Financial Cyber Crimes: Online banking fraud, credit card fraud, and other financial scams. 9. Data Theft: Stealing sensitive data, such as personal information or business data. 10. Malware Distribution: Spreading malicious software, such as viruses or ransomware.	4

SECTION – E		
36.	(i) print(sturec[3]) (ii) print(sturec[2][3]) (iii) print(max(sturec[2])) (iv) print(sturec.pop()) (v) sturec[0]=”Jai Dev”	5

37.	(a)YELLOW#RED#
	Maximum Value of R=2
	Minimum Value of R=1
	(b) A module is a chunk of Python code that exists in its own .py file and is intended to be used by Python code outside itself.
	The modules can be “imported” in other programs so the functions and other definitions in imported modules become available to code that imports them
	OR
	#to create a dictionary with five states and their capital in a dictionary and print its capital by entering a state
	print("to create a dictionary with five states and their capital in a dictionary and print its capital by entering a state")
	d=dict()
	for i in range(2):
	  st=input("enter the state:")
	  cp=input("enter its capital:")
	  d[st]=cp
	ist=input("enter the state to check its capital:")
	if ist in d:
	  print("capital of state ",ist," is ",d[ist])
	else:
	  print("capital of state ",ist," is not found in the dictionary")
		2+1
		= 3
		
		2

KENDRIYA VIDYALAYA SANGATHAN
SESSION ENDING EXAMINATION 2025-26
SAMPLE PAPER- UNSOLVED -1
Class: XI

Computer Science (083)

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

General Instructions:

- This question paper contains 37 questions.
- All questions are compulsory. However, internal choices have been provided in some questions. Attempt only one of the choices in such questions
- The paper is divided into 5 Sections- A, B, C, D and E.
- Section A consists of 21 questions (1 to 21). Each question carries 1 Mark.
- Section B consists of 7 questions (22 to 28). Each question carries 2 Marks.
- Section C consists of 3 questions (29 to 31). Each question carries 3 Marks.
- Section D consists of 4 questions (32 to 35). Each question carries 4 Marks.
- Section E consists of 2 questions (36 to 37). Each question carries 5 Marks.
- All programming questions are to be answered using Python Language only.
- In the case of MCQ, the text of the correct answer should also be written.

Section-A (21 x 1 = 21 Marks)		
1.	**State True or False** The expression ((10 - 2) / 4 + 3) * 2 == 10.0 evaluates to True in Python.	**1**
2.	Priya finds useful content for her assignment on a blog. She rewrites it in her own words but does not mention the original blog as a source. Which of the following best explains whether this is plagiarism and why? a) No, because she used her own words, and that makes it original. b) Yes, because even when paraphrased, using someone else's ideas without credit is plagiarism. c) No, because blogs are public and don't require citation. d) Yes, but only if the teacher finds out.	**1**
3.	Which of the following shows the correct order of precedence (from lowest to highest) for the given Python operators? Operators: + (addition), and (logical AND), ** (exponentiation), < (less than), = (assignment) a) = < + and ** b) and = < + ** c) = and < + ** d) < = and + **	**1**

4.	Consider the following Python code:
	`for i in range(1, 6):`
	`  if i == 4:`
	`  break`
	`else:`
	`  print("Completed without break")`
	What will be the output of the code and why?
	a) "Completed without break", because the loop completes without hitting a break statement
	b) No output, because the else block is only executed if the loop is not broken
	c) "Completed without break", because the loop runs 5 times
	d) The code will give a syntax error due to the else with the for loop
		1

5.	Predict output lst = [1, 2, 3, 4, 5, 6, 7, 8] lst.pop(3) lst.remove(6) del lst[1] print(lst) What will be the output of the code? a) [1, 3, 5, 7, 8] b) [1, 3, 5, 7] c) [1, 3, 5, 6, 7, 8] d) [1, 2, 5, 7, 8]	**1**
6.	Find the output of the following statement: "sees".strip('se') a) es b) '' c) sees d) None	**1**
7.	Expand the abbreviation ASCII.	**1**

8.	Which of the following statements will correctly create a tuple and what will be its output?
	A = tuple("abc") # Statement 1
	B = tuple(["abc"]) # Statement 2
	C = tuple(("a", "b", "c")) # Statement 3
	D = tuple(123) # Statement 4
	What is correct about the above statements?
	a) Statements 1, 2, and 3 create tuples; Statement 4 gives an error
	b) All four statements correctly create tuples
	c) Only Statement 1 gives an error; others are correct
	d) Statement 2 gives an error because tuple( ) needs individual elements
		1

9.	What will be the output of the following code? data = [1, 2, 3] result = data[3] a) TypeError b) ValueError c) IndexError d) No error, it will print 3	**1**
10.	What will be the output of the following code? import math result1, result2 = math.ceil(5.6), math.floor(5.6) print(result1,result2,sep=",") a) 5, 5 b) 6, 5 c) 5, 6 d) 6, 6	**1**
11.	Which of the following is a violation of Intellectual Property Rights (IPR)? a) Using open-source software with its license terms b) Sharing a copyrighted movie file on a public website without permission c) Creating a unique logo for your business d) Writing a research paper and citing your sources correctly	**1**
12.	Choose the correct option based on types of memory: a) Primary – Hard Disk, Secondary – ROM, Cache, Flash Drive b) Primary – Cache, ROM; Secondary – Flash Drive, Hard Disk c) Primary – Flash Drive, ROM; Secondary – Cache, Hard Disk d) Primary – ROM, Flash Drive; Secondary – Cache, Hard Disk	**1**
13	Which of the following is a type of malware that secretly monitors and records a user's activity? a) Trojan b) Adware c) Virus d) Keylogger	**1**

14.	Which of the following statement(s) would give an error after executing the following code?
	S = "Hello" # Statement 1
	print(S[1]) # Statement 2
	S[0] = "h" # Statement 3
	S += " World" # Statement 4
		1

	print(S) # Statement 5 a) Statements 2 and 3 b) Statements 3 and 4 c) Only Statement 3 d) Only Statement 4	
15.	Arrange the following memory units in ascending order (from smallest to largest): a) Bit, Nibble, Byte, Kilobyte (KB), Megabyte (MB), Gigabyte (GB), Terabyte (TB), Petabyte (PB) b)Bit, Nibble, Byte, Kilobyte (KB), Gigabyte (GB), Megabyte (MB), Petabyte (PB), Terabyte (TB) c) Byte, Nibble, Bit, Kilobyte (KB), Megabyte (MB), Gigabyte (GB), Terabyte (TB), Petabyte (PB) d)Nibble, Bit, Byte, Kilobyte (KB), Megabyte (MB), Gigabyte (GB), Petabyte (PB), Terabyte (TB)	**1**
16.	A flowchart is used to: A. Display the output only B. Visually represent the steps of an algorithm C. Write code in any programming language D. Design a web page layout	**1**
17	Which of the following data types are valid for use as keys in a Python dictionary? a) List, Tuple, Integer, String b) Integer, String, Tuple c) Integer, String, List d) List, Dictionary, Integer	**1**
18	Which of the following is a type of system software that manages the hardware resources of a computer? a) Web browser b) Operating system c) Spreadsheet d) Media player	**1**
19	Which of the following is an example of a Trojan? a) A virus that spreads through email attachments b) An adware that displays unwanted pop-up advertisements c)A tool that helps block spam emails d)A program that hides in the background and allows hackers to control your computer remotely	**1**
Q20 and 21 are ASSERTION AND REASON 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 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		
20.	**Assertion(A)**: In Python, the expression 5 = x will result in a syntax error. **Reason(R)**: In Python, variables cannot be used on the right-hand side of an assignment statement.	**1**
21.	**Assertion (A):** In Python, using from math import sqrt allows you to use sqrt(16) directly without prefixing it with math. **Reason (R):**The from statement imports only the specified function or name directly into the current namespace.	**1**
Section-B (7 x 2=14 Marks)		
22.	Answer the following: a) Name the gate that receives only one input signal b) Draw a logic circuit for the boolean expression A+B'C+A'C	**2**
23.	Discuss the two common tools for developing an algorithm **OR**	**2**

	Draw a flowchart to count and print the numbers from 1 to 10	

24.	Identify the **incorrect output(s)** of the following code.
	import random
	fruits = ['apple', 'banana', 'cherry', 'date', 'elderberry']
	for i in range(2):
	  index = random.randrange(1, 5, 2)
	  print(fruits[index], end=' ')
	(a) apple cherry (b) date date (c) banana date (d) banana banana
		2

25.	Sanvya created a program to manage her list of favorite colors. She wanted to insert a new color at a specific position and remove an existing one. Here's her code: colors = ['red', 'blue', 'green', 'yellow'] colors.insert['pink', 2] colors.remove('black') print(colors) When she runs the program, she gets an error and the list doesn't change as expected. Help her to correct the errors and rewrite the correct program. Also underline each correction made. **OR** Tanvi has a list of students' ages, and she wants to find out how many students are older than 18. Write a Python program that does the following: a) Given a list of student ages, count how many students are older than 18 b) Print the count.	**2**
26.	Identify the type of malware based on the following description: a) Software that displays unwanted advertisements and may track user activities. b)A malicious program that disguises itself as legitimate software to trick users into installing it.	**2**

27.	Predict the output of the following code:
	d = {"apple": 15, "banana": 8, "cherry": 19}
	for key in d:
	  for j in range(0, d[key] //7):
	  print(j, '#', end="")
	  print()
		2

28.	In the field of computer science, there is a significant gender gap. How can we encourage more girls to take an interest in computer science and technology? Discuss two ways that schools and colleges can promote gender equality in the field. **OR** E-waste is a growing problem worldwide. What are two environmental impacts of improper disposal of electronic waste (e-waste), and how can individuals help in e-waste management?	**2**
Section-C (3 x 3 = 9 Marks)		

29.	What is the output of the following code?
	a) lst = [1, 2, 3]
	  lst.insert(-1, 99)
	  print(lst)
	b) my_dict = {'x': 10, 'y': 20}
	  print('z' in my_dict)
	c) print(list({'one': 1, 'two': 2}.keys( )))
		3

30.	Write a Python program that takes two tuples of equal length and creates a new tuple by adding corresponding elements from both tuples. For example, if the input tuples are (1, 2, 3) and (4, 5, 6), the output should be (5, 7, 9). **OR** Write a Python program that swaps the first and last elements of a tuple. The program should return a new tuple with the swapped elements. if the tuple1 = (10, 20, 30, 40, 50) then the program should print (50, 20, 30, 40, 10)	**3**
31.	A software developer named Raj used an open-source code that was licensed under the GPL (General Public License) for a new commercial application without providing appropriate credit or distributing the modified source code. Later, a user of the software filed a complaint, alleging that Raj violated intellectual property rights. a) What type of violation of Intellectual Property Rights (IPR) has occurred in this case? b) Mention two ways Raj could have avoided this violation. c)What term refers to the legal protection given to creations of the mind, such as inventions, artistic works, and trademarks?	**3**
SECTION D (4 X 4 = 16 marks)		
32.	Write a program to find the sum of series $1 + \frac{x}{1!} + \frac{x^2}{2!} + \cdots$ OR Write a Python program to check whether a number is a Harshad number or not. Note: A Harshad number (or Niven number) is an integer that is divisible by the sum of its digits.	**4**
33.	Write a program to accept a sentence from the user and perform the following: • Count and display the number of **words that start with a vowel**. • Check whether the sentence contains the word **"Python"** (case-insensitive). • Display the total number of **alphabetic characters only** • Count and display the number of **words that are palindromes**	**4**
34.	Read the following case and answer the questions below: Raj is working on a science project using a computer. While typing, he suddenly loses power. When the computer restarts, he notices that the unsaved data is lost. He asks his teacher why it happened. The teacher explains that this is due to the nature of the memory where data was being temporarily stored while he was working. Based on this situation, answer the following: a. Which type of memory was Raj using that lost the data when power went off? b. Name one characteristic of this type of memory. c. How is it different from ROM? d. Give one example program that is typically stored permanently in ROM.	**4**
35.	Define the following terms in one or two sentences each: a. Phishing b. Copyright c. Social Media Etiquette d. Safe Browsing	**4**

	SECTION E (2 X 5 = 10 Marks)	
36.	Write a program to accept some integers in a list. Display the following: a) Sum of even numbers in the list b) count the numbers that are divisible by 5 c) all 3 digit numbers d) list in the reverse order e) list in which each number is displayed in reverse order eg: l=[123,43,56,32,788] sum of even numbers: 876 numbers divisible by 5: 0 three digit numbers: 123, 788 list in reverse order: [788, 32,56,43,123] list in which each number is reversed : [321,34,65,23,887] **OR** Write a name of the function for the following description: a. This function divides a string into a list of substrings based on a delimiter. b. This function removes all whitespace characters from both the beginning and the end of a string. c. This function converts all characters in a string to lowercase. d. This function inserts a common character in between every character of the string e. This function breaks the string on the basis of a delimiter provided and returns a tuple with exactly 3 elements.	**5**
37.	Write a Python program to create a dictionary that contains the name and salary of n employees where the **name of the employee** as the key and the salary as the value. The program must find the sum of salaries of all employees. Also display the names of the employees who is getting the highest and lowest salary OR Write a program in python to do the following: • Accept a list of integers • scans each element of the list and create a dictionary that stores number as key and its frequency in the list as the value • scans each element in the list and creates a dictionary that stores the number as a key and “even” or “odd” as the value • scans each element in the list and creates a dictionary that stores the numbers as key and no of digits as the value • Display all the three dictionaries for example of list l=[123,45,2356,45,123,45] The output must be: Frequency count: {123:2,45:3,2356:1} Even or odd:{123:”odd”,45:”odd”,2356:”even”} no of digits : {123:3,45:2,2346:4}	**5**

BIBLIOGRAPHY

1. **NCERT Textbook for Computer Science – Class XI**
   National Council of Educational Research and Training (NCERT)
   https://ncert.nic.in
2. **CBSE Curriculum and Guidelines for Class XI Computer Science (083)**
   Central Board of Secondary Education
   https://cbseacademic.nic.in
3. **Python Official Documentation**
   Python Software Foundation
   https://docs.python.org/3/

Additional Content

1. **Sample Papers**:
   https://drive.google.com/file/d/1nBHhk6oxe7_P56PAn791K0tbc6goZBfa/view

2. **Mind Maps:**

   https://drive.google.com/file/d/1LIgzVMgVY-it7hcpsks1iIBpdhQdTYHC/view