

Class 11<sup>th</sup>

PSEB

# COMPUTER SCIENCE

## Notebook & Practical File

Name \_\_\_\_\_

Roll No\_\_\_\_\_ Section\_\_\_\_\_

Session 20\_\_ to 20\_\_

### Syllabus of Subject

Number Systems

DBMS and SQL

Computer System Maintenance

Python Programming

Cyber Security And Ethics

### Time Table



**COMPUTER SCIENCE**  
**CLASS XI**  
**(Session 2025-26)**

**Learning Outcomes:**

Students should be able to:

- a) Develop basic computational thinking using Python
- b) Explain and use of data types, operators and control statements and other fundamental topics of Python
- c) Ability to understand the issues related to Cyber-Crimes and Safety, E-Waste Management
- d) Introduction to IT Act
- e) Ability to follow basic cyber ethics
- f) Understanding of Various types of Number Systems used by Computer Systems
- g) Ability to have understanding of various concepts of Databases
- h) Ability to use and work with SQL statements

**Unit-Wise Syllabus:**

**Theoretical Section**  
**Unit I**

**Number System:**

- **Number Systems:** Introduction, Decimal Number System, Binary Number System, Octal Number System, Hexadecimal Number System

**DBMS and SQL:**

- **Database concepts:** Introduction to Data, Information, Database and its needs, Traditional File System v/s Database System
- **Data Models:** Introduction, Hierarchical Data Model, Network Data Model, Relational Data Model
- **Introduction to DBMS:** DBMS, Components of DBMS
- **Relational Data Model:** Relation, Attribute, Tuple, Domain, Degree, Cardinality, Keys (Candidate Key, Primary Key, Alternate Key, Foreign Key)
- **SQL**

**Computer System Maintenance:**

- **Introduction:** Computer Maintenance and Security
- **Preventive Maintenance:** Basic Guidelines for Preventive Maintenance
- **Booting and Safe Mode Problems:** How to Start Windows in Safe Mode, How to Fix Your PC in Safe Mode-Scan for Malware, Run System Restore, Uninstall Recently Installed Software, Update Hardware Drivers, To check system crashes

- **Installation of Device Drivers:** Download the drivers manually, How to install the driver; Plug and Play Hardware Installation
- **Type of Ports:** Serial Port, Parallel Port, PS/2 Port, Universal Serial Bus (or USB) Port, VGA Port, Power Connector, Modem Port. Ethernet Port. Digital Video Interface, DVI port
- **PC Security Tools:** Importance of PC Security tools
- Software Update and Upgrade
- **Introduction to Windows Operating Systems:** Windows 10, Windows 8.1, Windows 7
- Installation of Python (GUI), IDEs
- **Control Panel:** Display Properties, Mouse and Keyboard, Date and Time, Devices and Printers, Regional Settings, Fonts
- **Utility Programs:** File Compression tools, Disk Defragmentation, Disk Clean-up, Backup and Restore
- **Shutting Down Options:** Switch User, Log Off, Lock Screen, Restart, Sleep, Hibernate, Shut Down or Turn Off

## Unit-II

### **Python Programming:**

- **Basic concepts for Python Programming:** Python Character Set, Python Tokens (Keyword, Identifier, Literal, Operator, Punctuator), Variables, Concept of L-Value and R-Value, Use of Comments
- **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 Operator, Augmented Assignment Operators, Identity Operators (is, is not), Membership Operators (in, not in)
- **Introduction:** Use of Indentation, Sequential Flow, Conditional and Iterative Flow control
- **Expressions, Statement, Type Conversion & Input/Output:** Precedence of Operators, Expression, Evaluation of Expression, Python Statement, Type Conversion (Explicit & Implicit Conversion), Accepting Data as Input from the Console and Displaying Output

### **Cyber Security and Ethics**

- **Cyber-Crime:** Definition, Hacking, Spying (Eavesdropping), Phishing and Fraud Emails, Ransomware, Preventing Cyber Crime

- **Cyber Safety:** Safely Browsing the Web, Identity Protection, Confidentiality, Cyber Trolls and Bullying
- **Safely Accessing Web Sites:** Adware, Malware, Viruses, Trojans, Phishing and Identity Verification
- **E-Waste Management:** Proper disposal of used Electronic Gadgets
- **IT Act:** Indian Information Technology Act (IT Act) 2000

## Practical Section

### Unit I

#### Python Programming:

- **Conditional Statements:** if, if-else, if-elif-else, Flowcharts, Simple Programs: e.g.: absolute value, largest of 2 and 3 numbers, finding divisibility of numbers etc
- **Iterative Statements:** for loop, Range Function, while loop, Flowcharts, break and continue statements, nested loops, Suggested Programs: Generating Pattern, Summation of series, Finding the Factorial of a positive number etc

#### 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 1234 123 12 1	A AB ABC ABCD ABCDE

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

- 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.

## **Project**

- The aim of the class project is to create something that is tangible and useful using Python. This should be done in groups of five to six students and should be started by students at least 3 months before the submission deadline. The aim here is to find a real-world problem that is worthwhile to solve.
- Students are encouraged to solve general and the school level problems, such as-create a calculator for general calculations, create a project to solve mathematical problems like area / perimeter of circle / square / rectangle etc., create a project to calculate the result of a student, create a project to calculate the net bill for sale with discount, create a database to store the student information etc. Students are also encouraged to use comments properly in the programs.
- To do some of these above mentioned project suggestions, some additional learning is required. This should be encouraged. Students should know how to teach themselves.
- The students should be sensitized to avoid plagiarism and violations of copyright issues while working on projects. Teachers should take necessary measures for this.

**A: Multiple Choice Questions:**

1. Which of the following is the smallest number in the hexadecimal system?  
A) 0  B) 1  C) A  D) F
2. What is the binary equivalent of the decimal number 15 ?  
A) 0111  B) 1111  C) 0101  D) 011
3. Which number system is most commonly used in everyday arithmetic?  
A) Binary  B) Decimal  C) Hexadecimal  D) Octal
4. Radix of Hexadecimal number system is  
A) 2  B) 5  C) 8  D) 16
5.  $(38CB)_{16}$  is an example of which number system is?  
A) Binary  B) Decimal  C) Hexadecimal  D) Octal

**B : Write True or False for the following statements:**

1. The Binary Number 1111 represents the decimal number 15. (**True**)
2. The Hexadecimal system use 2 as its base/radix. (**False**)
3. Octal word comes from Latin word Oct which means 8. (**True**)
4. The Binary Number System can represent any number using digits only 0 and 1. (**True**)
5. In the Octal Number System, the digit '9' is valid. (**False**)
6. In the Hexadecimal System, the letter 'B' represents the Decimal value 12. (**False**)
7. Roman Number System is and example of Positional Number System. (**False**)

**C : Short Answer Type Questions:**

**Q1.** What is the Number System? Write the name of basic categories of number System?

Ans: A number system is a method of representing numbers, providing a structured way to express quantities and perform arithmetic operations. It defines the symbols used and the rules for combining them. Each number system follows a specific set of rules for writing and calculating values.

There are two basic categories of number systems: **non-positional** and **positional**.

1. In a non-positional number system, the value of a symbol does not depend on its position (e.g., Roman numerals).
2. In a positional number system, the position of a digit affects its value, and this system is widely used in modern mathematics and computing. Common types of positional number systems include the **Decimal system** (base 10), **Binary system** (base 2), **Octal system** (base 8), and **Hexadecimal system** (base 16). These systems are especially important in fields like digital electronics and computer science, where binary and hexadecimal systems play a central role.

## Chapter-1

### Number System

Date \_\_\_\_\_  
Page No. \_\_\_\_\_

#### Q2. Explain Decimal Number System?

Ans: The **Decimal Number System** is the standard system used for representing numbers and is also known as the **Base-10 Number System**. The decimal system has a **base of 10**, meaning it uses **10 digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9**

#### Q3. What is binary Number System? Who is the inventor of this number?

Ans: The **Binary Number System** is a number system that uses only **two digits: 0 and 1**.

It is also known as the **Base-2** number system.

1. Each binary digit is called a **bit**.
2. It is mainly used in **computers and digital systems** because electronic devices can easily represent two states: **ON (1)** and **OFF (0)**.
3. Each place value in a binary number represents a power of 2.

#### Example:

The binary number **1011** represents:

$$1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 8 + 0 + 2 + 1 = 11 \text{ (in decimal)}$$



The binary number system was **formally developed by** the German mathematician and philosopher **Gottfried Wilhelm Leibniz** in **1703**.

He published a paper titled "*Explication de l'Arithmétique Binaire*", explaining how all numbers can be represented using just 0s and 1s.

#### Q 4. Explain Octal Number System?

Ans: The **Octal Number System** is a **base-8** number system that uses **eight digits: 0, 1, 2, 3, 4, 5, 6, and 7**. Octal is **Positional System**, Like decimal and binary, the value of each digit depends on its position and the base. Octal is often used in **computing**, especially with **older systems or Unix file permissions**. **For Example:**

#### Unix/Linux File Permissions

- Each permission (read, write, execute) is represented by a bit.
- Example:
  - chmod 755 file.txt sets:
    - Owner: read (4) + write (2) + execute (1) = 7
    - Group: read (4) + execute (1) = 5
    - Others: read (4) + execute (1) = 5
  - Octal provides a **compact way** to represent binary permission flags.

# Chapter-1

## Number System

Date \_\_\_\_\_  
Page No. \_\_\_\_\_

### Q5. Explain Hexadecimal Number System?

Ans: The **Hexadecimal Number System** is a **base-16** system that uses **16 symbols**: **0–9** (representing values 0 to 9) and **A–F** (representing values 10 to 15). Each digit in hexadecimal corresponds to **four binary bits**, making it very useful in computing.

#### Practical Uses:

- Memory Addresses:** Computers use hexadecimal to represent memory locations because it's shorter and more readable than binary.
- Color Codes in Web Design:** Colors in HTML/CSS are often defined using hexadecimal (e.g., #FF5733 = Red: FF, Green: 57, Blue: 33 )

### D : Perform the following conversion of Number System

$$1. \quad (25)_{10} = (11001)_2 = (31)_8 = (19)_{16}$$

2	25	1	LSB	8	25	1	16	25	9
2	12	0			3			1	
2	6	0							
2	3	1							
1			MSB						

$$2. \quad (10111)_2 = (23)_{10} = (27)_8 = (17)_{16}$$

1	0	1	1	1	0	1	0	1	1	0	0	0	1	0	1	1	1	
2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
16	8	4	2	1	4	2	1	4	2	1	8	4	2	1	8	4	2	1
1x16	0x8	1x4	1x2	1x1	0x4+1x2+0x1	1x4 + 1x2 +1x1	0x4+1x2+0x1	0x8+0x4+0x2+1x1	1x4 + 1x2 +1x1	0x8+1x4+1x2+1x1								
16 + 0	+ 4	+ 2	+ 1															
23					2		7			1					7			

$$3. \quad (47)_8 = (39)_{10} = (100111)_2 = (27)_{16}$$

4	7	4	7	0	0	1	0	0	1	1	1
8 <sup>1</sup>	8 <sup>0</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
8	1	4	2	1	4	2	1	8	4	2	1
4x8	7x1	(1x4+0x2+0x1)	(1x4 + 1x2 +1x1)	(0x8+0x4+1x2+0x1)	(0x8+1x4+1x2+1x1)						
32 + 7 = 39		1 0 0	1 1 1	2				7			



# Chapter-1

## Number System

Date \_\_\_\_\_  
Page No. \_\_\_\_\_

$$4. \quad (A4)_{16} = (164)_{10} = (1010\ 0100)_2 = (244)_8$$

A	4	A	4	0	1	0	1	0	0	1	0	0
$16^1$	$16^0$	$2^3$	$2^2$	$2^1$	$2^0$	$2^3$	$2^2$	$2^1$	$2^0$	$2^2$	$2^1$	$2^0$
16	1	8	4	2	1	8	4	2	1	4	2	1
$10 \times 16$	$4 \times 1$	$1 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 1$	$0 \times 8 + 1 \times 4 + 0 \times 2 + 0 \times 1$			$0 \times 4 + 1 \times 2 + 0 \times 1$			$1 \times 4 + 0 \times 2 + 0 \times 1$		$1 \times 4 + 0 \times 2 + 0 \times 1$	
$160 + 4 = 164$		1010		0100			2		4		4	

### Activity

#### 1.1 Complete the Following Table

Decimal	Binary	Hexadecimal	Octal	
0	0000	0	0	
1	0001	1	1	
2	0010	2	2	
3	0011	3	3	
4	0100	4	4	
5	0101	5	5	
6	0110	6	6	
7	0111	7	7	
8	1000	8	10	
9	1001	9	11	
10	1010	A	12	
11	1011	B	13	
12	1100	C	14	
13	1101	D	15	
14	1110	E	16	
15	1111	F	17	

#### 1.2 Perform the Following Conversions

Decimal	Binary	Octal	Hexadecimal	
23	0001 0111	27	17	
61	111101	75	3D	
23	0001 0111	27	17	
908	0011 1000 1100	1614	38C	
89	0101 1001	131	59	
17	10001	21	11	
143	1000 1111	217	8F	
31	0001 1111	37	1F	
154	1001 1010	232	9A	
99	0110 0011	143	63	



**A: Multiple Choice Questions:**

1. The \_IDLE\_ is a default editor that accompanies Python.  
 A) IDLE  B) IPLE  C) Text Editor  D) Notepad
2. The >>> is the shell prompt where we type in our commands.  
 A) <<  B) >>  C) >>>  D) <<<
3. IDE stands for **Integrated Development Environment**.  
 A) Integrated Direct Environment   
 B) **Integrated Development Environment**   
 C) Information Development Environment   
 D) None of These
4. To run a script file code in Python, we can use F5 shortcut key.  
 A) F1  B) F2  C) F5  D) F7
5. Tokens are like words and punctuation marks in English language.  
 A) Literals  B) Identifiers  C) Variables  D) Tokens
6. Identifiers are the names given to program elements such as variables, functions, lists, tuples, classes, etc., to uniquely identify them in code.  
 A) Literals  B) Identifiers  C) Variables  D) Tokens
7. Literals are the Fixed values used in a source code.  
 A) Fixed  B) Boolean  C) String  D) Float
8. Variables are those identifiers which are used to store values and they allow us to change their value during runtime.  
 A) Constant  B) Variable  C) List  D) Tokens
9. A Comment is basically a text that gives and Explanation about the program code.  
 A) Execution  B) Compilation  C) **Explanation**  D) All of these
10. We can display program data to the console in Python with print() function.  
 A) Input()  B) **print()**  C) output()  D) All of these

**B : Short Answer Type Questions:**

**Q1.** What do you know about Python?

**Ans:** Python is a high-level, interpreted programming language known for its simple and easy-to-read syntax. It supports multiple programming paradigms like procedural, object-oriented, and functional programming. Python is widely used in web development, data science, machine learning, automation, and more. It was created by **Guido van Rossum** and first released in 1991.

## Chapter-2

### Basic Concepts of Python Programming

Date \_\_\_\_\_  
Page No. \_\_\_\_\_

#### Q2. What is IDE?

Ans: An IDE (Integrated Development Environment) in Python is a software application that provides comprehensive facilities for Python programming. It includes features like code editing, debugging, running Python code, and managing projects. Popular Python IDEs include **IDLE**, **PyCharm**, **VS Code**, and **Jupyter Notebook**. They make coding easier by offering syntax highlighting, autocompletion, and other helpful tools.

#### Q3. How can you view the list of Keywords using Python Shell? Write Some examples of Keywords.

Ans: Python has a set of **reserved keywords** that are predefined and have special meanings. These keywords cannot be used as identifiers (such as variable names or function names) because they are used by the Python language itself.

You can view the list of keywords in Python by using the keyword module. In Python Shell, you can execute the following code:

```
import keyword  
print(keyword.kwlist)
```

This will display a list of all the reserved keywords in Python's are: *False, await, else, import, pass, None, break, except, in, raise, True, class, finally, is, return, and, continue, for, lambda, try, as, def, from, nonlocal, while, assert, del, global, not, with, async, elif, if, or, yield*

#### Q4. How will you declare variable in Python? Give Examples

Ans: In Python, variables are declared by simply assigning a value to a name. No explicit declaration is needed, and Python automatically determines the variable's type based on the assigned value.

```
x = 5          # Integer  
name = "Balli " # String  
is_active = True # Boolean
```

Here, x is assigned the value 5 (integer), name is assigned "balli " (string), and is\_active is assigned True (boolean).

#### Q5. Why print() function is used in Python programs?

Ans: The print() function in Python is used to output data to the console or terminal. It allows you to display values, variables, or messages for the user to see. This is helpful for debugging, showing results, or providing information to the user.

```
print("Hello, World!") # This will print the string "Hello, World!" on the screen.
```

#### C : Long Answer Type Questions:

Q1. Explain Token ? Explain Various types of tokens used in Python.

Ans: A **token** is the smallest unit of a program that has meaningful representation in Python. In simpler terms, tokens are the building blocks of a Python program. When you write Python code, the interpreter breaks it down into these individual tokens for further processing. Tokens can be classified into several categories, each representing a specific type of element in the program.

The main types of tokens in Python are:

1. **Keywords:** These are reserved words with predefined meanings that cannot be used as identifiers. They define the structure of the language, such as control flow and function definitions.
2. **Identifiers:** These are names used to identify variables, functions, classes, and other objects in a program. They are defined by the programmer.
3. **Literals:** These represent constant values directly in the code, such as numbers, strings, and boolean values. They are used to assign values to variables.
4. **Operators:** Symbols that perform operations on variables and values, including arithmetic, comparison, logical, and assignment operations.
5. **Punctuation (Delimiters):** These include symbols such as parentheses, brackets, commas, and colons, which structure the code and separate different elements.
6. **Comments:** These are used for adding explanations or notes within the code and are not executed.

Q2. What are Identifiers ? write down their naming rules.

Ans: In Python, **identifiers** are names used to identify variables, functions, classes, modules, or other objects. An identifier allows the program to access and manipulate the value associated with it. It is essentially a label used to reference data in the program.

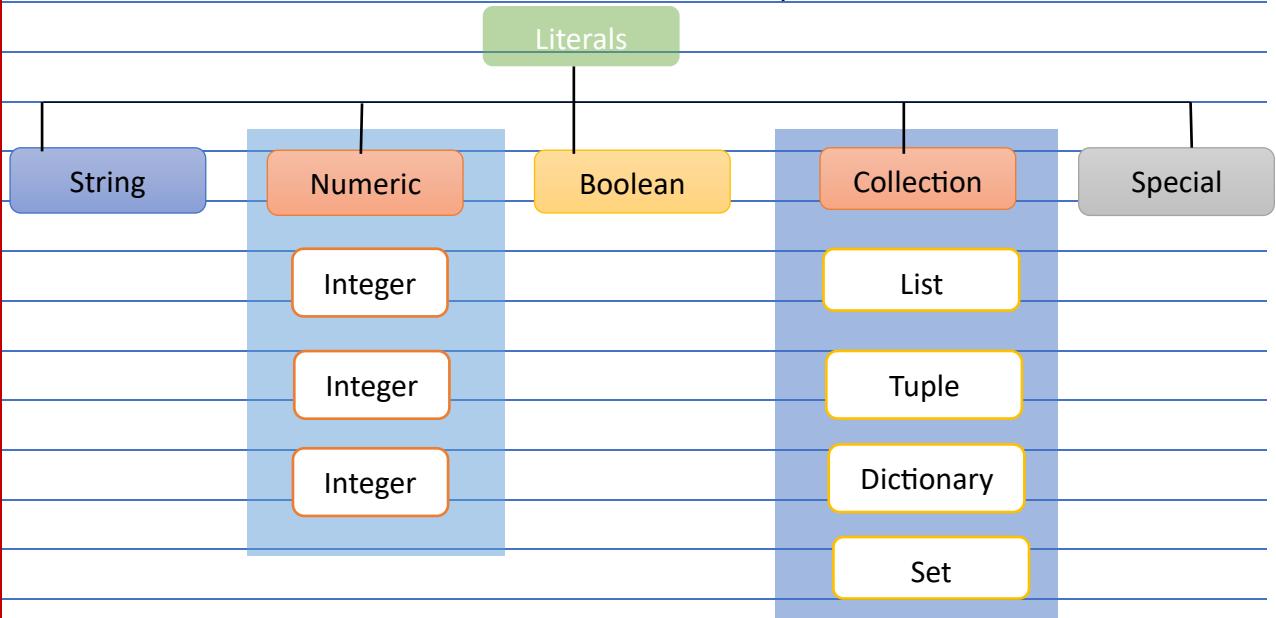
**Naming Rules for Identifiers in Python:**

1. **Start with a Letter or Underscore:** An identifier must begin with a letter (a-z, A-Z) or an underscore (\_). It cannot start with a digit (0-9).
2. **Subsequent Characters:** After the first character, an identifier can contain letters, digits (0-9), and underscores (\_).
3. **Case Sensitivity:** Identifiers are case-sensitive, meaning myvariable, MyVariable, and MYVARIABLE are considered different identifiers.
4. **No Reserved Keywords:** Identifiers cannot be Python keywords (such as if, else, True, False, etc.), as these have special meanings in Python.
5. **No Special Characters:** Identifiers cannot contain spaces or special characters (e.g., @, #, &, \*, etc.).
6. **Length:** There is no fixed length for identifiers, but it is recommended to keep them descriptive yet concise for readability.

**Q3.** What are Literals? Draw a chart to represents the different types of literals used in Python.

Ans: In Python, **literals** are fixed values or raw data that appear directly in your code. They represent constant values of built-in data types. When you write a literal, you are essentially telling Python the exact value you want to use.

For example, 10 is an integer literal, "hello" is a string literal, 3.14 is a floating-point literal, and True is a boolean literal. These values are embedded directly into the source code.



**Q4.** What are Comments? Write the Different Ways of Writing Comments in Python.

Ans: Comments are explanatory notes added to your code that are ignored by the Python interpreter. They are used to make your code more readable and understandable for yourself and others.

Good comments explain the "why" behind the code, not just the "what."

Here are the different ways to write comments in Python:

#### 1. Single-line Comments:

You can write a single-line comment by starting the line with a hash symbol (#). Anything after the # on that line is considered a comment and will not be executed.

`x = 10 # Assigning the value 10 to the variable x.`

#### 2. Multi-line Comments (Using Triple Quotes):

While Python doesn't have a specific syntax for multi-line comments like some other languages (e.g., /\* ... \*/ in C++ or Java), the common and widely accepted way to write multi-line comments is by enclosing the text within triple quotes (" or """). These are technically multi-line strings, but when they are not assigned to a variable, they are treated as comments by the interpreter.

`""" You can also use double  
triple quotes for  
multi-line comments. """`

#### A: Multiple Choice Questions:

1. Which of the following program element is used to hold value in memory?  
**A) Variable**       B) Comments      C) Operators      D) All of these
2. Which of the following is not a Standard Numeric Data type in Python?  
A) Integer      B) Floating      **C) Boolean**       D) Complex
3. Which of the following represents the Mapping Data type in Python?  
A) List      **B) Dictionary**       C) Tuple      D) Set
4. The \_\_\_\_\_ keyword is used to define a null value, or no value in Python?  
A) Nothing      B) Null      C) Zero      **D) None**
5. The \_\_\_\_\_ Keyword is used to represent the True/ False value in Python.  
A) **bool**       B) boolean      C) Boolean      D) None
6. For Creating a list a Python, we write data items in \_\_\_\_\_  
A) Parenthesis      **B) Square brackets**       C) Curley Brackets      D) Angular Brackets
7. \_\_\_\_\_ in python are used to represent Unicode character values.  
A) **String**       B) Tuple      C) List      D) Set
8. \_\_\_\_\_ types are those types whose contents are allowed to be changed after the creation.  
A) **Mutable**       B) Immutable      C) None      D) Mapping
9. Which of the following is not an Arithmetic Operator in Python?  
A) /      B) //      C) \*\*      **D) ++**
10. Which of the following function(s) is used for type conversion?  
A) int()      B) float()      C) str()      **D) all of these**

#### B: Write True or False for the Following statements:

1. Symbol % represents modulus operator which is used to get the remainder value after division of two numbers.  **True**
2. Relational operators return either True or False value After Comparison.  **True**
3. Operators are the special symbols which are used to perform common operations on operands.  
 **True**
4. A set is an unordered collection of comma-separated values in the square brackets.  **False**
5. A Sequence is an ordered collection of similar or different data items.  **True**

### C: Short Answer Type Questions:

Q1. What are floating point numbers. Give Examples.

Ans: Floating point numbers are numbers that have a decimal point or are written in exponential (scientific) notation. They are used to represent non-integer (real) values, including very large or very small numbers with fractional parts. In Python (and many other languages), float is the data type used for floating point numbers. Examples: 3.14, 0.5, 1.2e3 (which means  $1.2 \times 10^3 = 1200.0$ )

Q2. What do you know about the Boolean Data Type of Python?

Ans: The Boolean data type (Booleans are a subtype of integers, Boolean values are case-sensitive) in Python represents one of two possible values:

1. True is treated as 1

2. False is treated as 0

These values are used to represent the truth value of expressions and are commonly used in conditions, comparisons, and logical operations.

Q3. What are the sequence Data Type in Python? Write their Name.

Ans: The sequence data types in Python are those that can hold multiple items in an ordered manner, meaning the position of each item matters. Here are their names:

1. List

2. Tuple

3. Range

4. String (also a sequence of characters)

5. Byte (for binary data)

6. Bytearray (mutable version of bytes)

Q4. Explain Operands with suitable examples.

Ans: An **operand** is a value or variable on which an operator performs an operation. In expressions, operators act on operands to produce a result. For Example:

a = 10

b = 5

result = a + b # a and b are operands, + is the operator

print(result) # Output: 15

### Chapter-3

## Data Types, Operation & Expressions Python Programming

Date \_\_\_\_\_  
Page No. \_\_\_\_\_

#### Q5. What are Arithmetic Operators? Give Examples.

Ans: In Python, Arithmetic operators are symbols used to perform mathematical operations on numerical values, known as operands. For Example: **Arithmetic Operators: +, -, \*, /, %, //, \*\***

```
a = 10 # a is operands  
b = 3 # b is operands  
print(a + b) # 13  
print(a - b) # 7  
print(a * b) # 30  
print(a / b) # 3.333...  
print(a % b) # 1  
print(a // b) # 3  
print(a ** b) # 1000
```

#### Q6. What is Expression?

Ans: An expression in Python is a combination of values, variables, operators, and function calls that Python interprets and evaluates to produce a result.

```
result = 5 + 3
```

# Expression: 5 + 3 → Evaluates to 8

#### D: Long Answer Type Questions:

#### Q1. What do you mean by Data Types? Draw a chart representing the various common standard data types used in Python.

Ans: **Data types** in Python define the kind of value a variable can hold and what operations can be performed on that value. For example, whether it's a number, text, list, or something else.

#### Examples:

- int: a = 10
- float: b = 3.14
- str: name = "balli"
- list: numbers = [1, 2, 3]
- dict: info = {"name": "balli", "age": 25}

#### Python DataTypes

##### Numeric

Integer x = 10  
float x = 3.14  
Complex x = 1 + 2j

##### Boolean

True, False

##### Set

x = {1, 2, 3}

##### Sequence

String "Hello, World!"  
Tuple Ordered, Immutable collection  
x = (1, 2, 3)  
List Ordered, mutable collection  
x = [1, 2, 3]  
Range Sequence of numbers  
x = range(5)

##### Mapping

Dictionary (Key-value pairs)  
x = {"name": "Alice", "age": 25}

##### None



**Q2. What is the difference between Mutable and Immutable Types?**

Ans: In Python, **Mutable** refers to **objects that can be changed after they are created**. That means you can modify their **content, size, or structure** without changing their identity (i.e. the object stays the same in memory). **For Example** : list, dict, set and bytearray

```
numbers = [1, 2, 3]
```

```
numbers[0] = 100 # Modifies the original list
```

```
numbers.append(4) # Adds new item
```

```
print(numbers) # Output: [100, 2, 3, 4]
```

An **immutable** object in Python is **one that cannot be changed** after it is created. If you try to modify it, a new object is created instead. **For Example** : int, float, str, tuple, bool, frozenset and bytes

```
x = 5
```

```
x = x + 1 # A new integer object is created
```

```
print(x) # Output: 6
```

**Q3. What are Operators? Explain any two types of operators used in Python.**

Ans: **Operators** in Python are **symbols or keywords** used to perform operations on variables and values. They are the core of all computations, allowing actions like addition, comparison, logic testing, and more.

Python supports several types of operators, including:

- **Arithmetic operators**
- **Comparison (relational) operators**
- **Assignment operators**
- **Logical operators**
- **Bitwise operators**
- **Membership operators**
- **Identity operators**

**Arithmetic Operators:** Used to perform mathematical operations like addition, subtraction, multiplication, etc. . For Example: **Arithmetic Operators: +, -, \*, /, %, //, \*\***

```
a = 10 , b = 3
```

```
print(a + b) # Output: 13
```

```
print(a % b) # Output: 1
```

**Comparison Operators:** Used to **compare two values**. They return True or False. For example

```
x = 10 , y = 20 ,
```

```
print(x < y) # Output: True
```

```
print(x == y) # Output: False
```

### Chapter-3

## Data Types, Operation & Expressions

### Python Programming

Date \_\_\_\_\_

Page No. \_\_\_\_\_

Q3.2 Try find the Answer for the following Statements:

1. Which of the following statement is not correct ?

- A) `print("7"+9")`    B) `print(int("seven"))`     C) `print(7+9)`    D) `print(7)`

**Explanation: You can't convert the string "seven" to an integer. It will raise a ValueError.**

2. Which of the Following is not a correct Arithmetical Expressions?

- A) `A=3-1*2`    B) `Y+=1`    C) `15+6=Z`     D) `Ten=5*4`

**Explanation: In Python, assignment must be written as `Z = 15 + 6`. The expression `15 + 6 = Z` is a syntax error.**

3. Which of the following is correct example of type conversion ?

- A) `Int(5.0+"0.1")`    B) `int("four")`    C) `int("77"+"five")`    D) `int(5.0+2.3)`

**Explanation:  $5.0 + 2.3$  gives  $7.3$ , and `int(7.3)` results in  $7$ . It's a valid conversion.**

4. Which of the following expression does not produce 4 result ?

- A) `17//4`    B) `13%9`    C) `2**9`     D) `17/4`

**Explanation:  $2^{**}9 = 512$ , not 4. Others:**

- $17//4 = 4$
- $13 \% 9 = 4$
- $17 / 4 = 4.25$  (but close, though technically not exactly 4)

**So if "does not produce 4" means exactly not 4, C and D are both technically correct. But C is the most clearly incorrect.**

5. Which of the following is a wrong way of comments in Python ?

- A) `"""this is comment """`    B) `# this is Comment`  
C) `//this is comment`     D) `##### This is Comment`

**Explanation: // is not a comment symbol in Python (it is in languages like C++ and Java).**

**Python uses # or triple quotes for multiline strings.**

6. Which of the following are the correct values of Boolean data type in Python ?

- A) Yes/No    B) Agree/Disagree    C) True/False     D) Right/Wrong

**Explanation: True and False are the only valid Boolean values in Python.**

7. If the Value of `a=20` and `b=20`, then `a+=b` will assign \_\_\_\_\_ to `a`

- A) 40     B) 30    C) 20    D) 10

**Explanation: `a += b` is equivalent to `a = a + b`, so  $20 + 20 = 40$**

8. The \_\_\_\_\_ operator is used to find out if division of two number yield any remainder.

- A) /    B) +    C) %     D) //

**Explanation: The modulus operator % returns the remainder of a division.**

### Chapter-3

## Data Types, Operation & Expressions Python Programming

Date \_\_\_\_\_

Page No. \_\_\_\_\_

Q3.3 Execute the following statements in Python IDLE and write down the outputs :

A) `print("Hello" + "Friends")`

**Explanation:** String concatenation using +

HelloFriends

B) `print(15+7-3*2%4)`

**Explanation:**

$$3 * 2 = 6$$

$$6 \% 4 = 2$$

$$15 + 7 - 2 = 20$$

20

C) `a=5.2 #float Value`

```
msg="Good Morning" # String Value
c=4+3j #complex value
d=5%2.0 #Modulus Operator
print(msg+"Friends")
print(msg,"Friends")
print(a,c,5/2.0,5//2.0,d)
```

"Good Morning" + "Friends" → concatenated string2

5 / 2.0 → float division → 2.5

5 // 2.0 → floor division → 2.0

5 % 2.0 → modulus → 1.0

4 + 3j is a complex number

Good MorningFriends

Good Morning Friends

5.2 (4+3j) 2.5 2.0 1.0

D) `x,y=12,6`

`x,y,y=x+2`

`print(x,y)`

**Explanation:**

- $x = 12, y = 6$

- After  $x, y = y, x + 2$ :

- $x = 6$  (old value of y)

- $y = 12 + 2 = 14$

6 14

E) `p,q,r=5,2,1`

`p**=q+r`

`r='5'+'5'`

`print(p,q,r)`

**Explanation:**

- $q + r = 2 + 1 = 3$

- $p **= 3 \rightarrow 5 ** 3 = 125$

- $r = '5' + '5' \rightarrow$  string concatenation → '55'

125 2 55

F) `result=float(10)+int(5.2)`

`print(result)`

**Explanation:**

- $\text{float}(10) = 10.0$

- $\text{int}(5.2) = 5$

- $10.0 + 5 = 15.0$

15.0

G) `a=7`

`b=4`

`exp1=(a>b)`

`print("Result of Expression: ",exp1)`

Result of Expression: True

**Explanation:**

- $a > b \rightarrow 7 > 4 \rightarrow \text{True}$



**Q3.4** Find the Errors in the following code and write the Correct Statements:

A) `print 7`

**Error:** Missing parentheses.

In Python 3, print is a function and must be used with parentheses.

`print(7)`

B) `print(Hello)`

**Error:** Hello is treated as a variable because it's not in quotes, but it's not defined.

`print("Hello")`

C) `print("Hello"+Friends)`

**Errors:** Friends is not quoted and treated as an undefined variable.

`print("Hello" + "Friends")`

D) `a==10`

`print("Value of a is " a)`

**Errors:**

- `a == 10` is a comparison, but assignment is needed (`a = 10`).
- The print statement is missing a comma or + between string and variable.

`a = 10`  
`print("Value of a is", a)`

### A: Multiple Choice Questions:

1. If we have to select one from two or more options during programming, then we will use which of the following options:  
A) Simple if    B) **ifelif else**     C) Sequence Execution    D) None of these
2. How will you define the logical expression for the following condition?  
"Marks are greater than or equal to 60 but less than 80"  
A) If marks  $\geq 80$  and marks  $\leq 60$     C) If marks  $> 60$  and marks  $\leq 80$   
B) If marks  $\geq 60$  and marks  $\leq 80$     D) **If marks  $\geq 60$  and marks  $< 80$**
3. When we use if else condition within another if else condition, it is called?  
A) else if ladder    B) Simple if else    C) **Nested if else**     D) None of these
4. Conditional Flow Control is also known as?  
A) Branching Statement    C) Looping Statement  
B) Decision Making Statement    D) **Both a and b**
5. Looping Statement is also known as?  
A) **Iterative Statement**     C) Sequential Statement  
B) Conditional Statement    D) All of these
6. If we have to repeat the execution of one or more statements, we can use  
A) **Iterative Statement**     C) Conditional Statements  
B) Skipping Statements    D) None of these
7. How will you define the range function for generating following sequence?  
"2,4,6,8,10,12,14,16"  
A) range(2,16)    C) range(2,17)  
B) **range(2,17,2)**     D) range(2,16,2)
8. When we use a loop within another loop, it is called ?  
A) Infinite Loop    B) **Nested Loop**   
B) for while loop    D) None of these
9. A loop that never ends is called a:  
A) Continuous Loop    C) **Infinite Loop**   
B) Circular Loop    D) None of these
10. The \_\_\_\_\_ statements in Python brings control out of loop.  
A) **Break**     B) back    C) pass    D) continue
11. Which of the following loop will continue infinitely ?  
A) while 0:    B) **while 1:**     C) while:1:    D) while False:

## Chapter-4: Control Statements

C: Complete/ Modify the Following Code :

```
num1 = int(input("Enter first number: "))

num2 = int(input("Enter second number: "))

if num1 == num2:
    print("Both numbers are equal")
elif num1 > num2:
    print("num1 is greater than num2")
else:
    print("num2 is greater than num1")
```

D: Write the Output of following Code and do changes as specified below:

```
num1 = 10
num2 = 5

if num1 > num2:
    if(num1%num2==0):
        print(num1, "num1 is Divisible by", num2)
    else:
        print(num1, "num1 is not Divisible by", num2)
else:
    print(num1*num2)
```

Ans. **Output Explanation:**

- num1 = 10, num2 = 5
- 10 > 5 → True
- 10 % 5 == 0 → True (10 is divisible by 5)

**Output:**

**10 num1 is Divisible by 5**

What changes will you do in above code to execute print(num1\*num2) statement?

When the statement “if num1 > num2:” becomes false, the else block is executed, and the print(num1 \* num2) statement runs. For example

```
num1 = 3
num2 = 5
```

Now, 3 > 5 is False, so it enters the else block.

### E: Write the Output of following Code

#### 1. Decrementing num using while loop:

Num=10

While num> 1:

    print(num)

    num-=2

10  
8  
6  
4  
2

#### 5. Factorial using while loop

i=1;f=1;n=5

while i <=n:

    f= f\*i

    i=i+1

120

print(f)

#### 2. Multiples of 5 using for loop

For i in range(1,11):

    print(i\*5)

5  
10  
15  
20  
25  
30  
35  
40  
45  
50

#### 6. Print a right-angled triangle pattern of asterisks using nested for loops

for row in range(1,5):

    for col in range(1,row):

        print("\*", end="")

\*  
\*\*  
\*\*\*

print()

#### 3. Counting Down

for r in range(15, 0, -2):

    print(r)

15  
13  
11  
9  
7  
5  
3  
1

#### 4. Iterating over a string

State='PUNJAB'

for s in state:

    print(s)

P  
U  
N  
J  
A  
B

## Chapter-5: Basic Concepts of Database Management System

Date \_\_\_\_\_  
Page No. \_\_\_\_\_

### A: Multiple Choice Questions:

1. Which Network Model is designed in the form of plex network?

- A) Hierarchical Model      C) Network Model

- B) Relational model      D) None of these

**Explanation:** The Network Model allows complex relationships with many-to-many connections and is often visualized as a plex structure (graph-like).

2. Which of the following represents the name of Column in a table?

- A) Relational      B) Attribute       C) Degree      D) tuple

**Explanation:** In relational databases, columns are referred to as attributes.

3. Which of the following is not a component of Database?

- A) Hardware      B) Software      C) Network       D) User

**Explanation:** Database components include hardware, software, users, and data. Network is part of infrastructure but not a core database component.

4. Who is responsible to administrate the database?

- A) End Use    B) System Programmers    C) Database Administrator     D) Data Modelers

**Explanation:** A Database Administrator (DBA) manages, maintains, and secures the database.

5. An organized or classified Data is known as:

- A) Information     B) Table      C) Result      D) None of these

**Explanation:** When data is processed, organized, or structured, it becomes information.

### B: Fill in the Blanks:

1. File-based systems were traditional systems for the storage of facts in a manual way.

2. Schema can be referred as logical structure of the Database.

3. The total number of records present in table is called cardinality of a relation.

4. The primary key refers to identify all the records uniquely in a particular column.

5. SQL (Structured Query Language) is a standardized language which provides commands to define the storage groups, different structures and objects in a database.

### C: Write the full forms of the following:

1. RDBMS – Relational Database Management System

2. DBMS – Database Management System

3. DML – Data Manipulation Language

4. DBA - Database Administrator

5. DDL – Data Definition Language

6. SQL – Structured Query Language

### D: Short Answer Type Questions:

Q1. Define Data and Information.

Ans: 1. **Data**: Raw, unorganized facts or figures without context. Data can be numbers, text, images, sounds, etc., and by itself, it may not have meaning.

**Example**: "100", "blue", "John", "23".

2. **Information**: Processed, organized, or structured data that has meaning and is useful for decision-making or understanding.

**Example**: "John is 23 years old and has blue eyes."

Q2. What do you mean by DBMS.

Ans: DBMS stands for **Database Management System**, It is a **software system** that allows users to **create, store, manage, retrieve, and update** data in a **structured** and **efficient** way. A DBMS is like a **digital filing cabinet** that helps you organize and manage lots of information easily and securely.

Q3. Explain the Hardware Component of the Database.

Ans: The **hardware component** of a database refers to the **physical devices** required to run the database system and store/manage data. These are the foundational parts that support the functioning of the DBMS. The hardware ensures the **performance, availability, and reliability** of a database system. Without appropriate hardware, even the best DBMS software cannot function efficiently.

Q4. Write a short note on Data Models.

Ans: A **data model** is a **conceptual framework** that defines how data is **organized, stored, and related** within a database. It provides a **blueprint** for designing a database structure and helps in understanding the data flow.

#### Types of Data Models:

1. **Hierarchical Model**
2. **Network Model**
3. **Relational Model**
4. **Object-Oriented Model**
5. **Entity-Relationship (E-R) Model**

Q5. Define the relation in Database.

Ans: In database terminology, a **relation** refers to a **table** in a **relational database**. It is a structured format to store data in **rows** and **columns**. A **relation** is simply a **table** in a relational database that organizes data in a meaningful, structured way.

Each **row** (also called a **tuple**) represents a single record.

Each **column** (also called an **attribute**) represents a field of the record.

**Q6. What do you mean by degree and cardinality of relations ?**

Ans: In relational databases, **degree** and **cardinality** describe the **structure** and **size** of a table (relation).

### **1. Degree of a Relation**

- **Definition:** The **number of attributes (columns)** in a relation.
- It indicates how many fields each record contains.

**Example:**

ID	Name	Age
1	John	22

This table has **3 columns** → So, **degree = 3**

### **2. Cardinality of a Relation**

- **Definition:** The **number of tuples (rows)** in a relation.
- It shows how many records are stored in the table.

**Example:**

ID	Name	Age
1	John	22
2	Mary	23
3	Alex	21

This table has **3 rows** → So, **cardinality = 3**

**Q7. Define the Primary Key of a database.**

Ans: A **Primary Key** is a **special attribute or a set of attributes** in a database table that **uniquely identifies each record (row)** in that table.

1. **Unique:** No two rows can have the same primary key value.
2. **Not Null:** A primary key cannot contain null (empty) values.
3. Each table can have **only one primary key**.

**Example:**

StudentID	Name	Age
101	Alice	20
102	Bob	21
103	Clara	22

- Here, **StudentID** is the **primary key** because it uniquely identifies each student.

## Chapter-5: Basic Concepts of Database Management System

Date \_\_\_\_\_  
Page No. \_\_\_\_\_

### D: Long Answer Type Questions:

Q1. What are the components of a Database? Explain in detail.

Ans: A **database** is a structured system for storing, managing, and retrieving data. To function effectively, it relies on several key components. These components work together to ensure the database is **efficient, secure, and reliable**.

Component	Description
<b>Hardware</b>	Physical equipment that runs the database
<b>Software</b>	DBMS and related programs
<b>Data</b>	Actual stored information
<b>Procedures</b>	Guidelines for managing and using the database
<b>Database Language</b>	SQL or similar language for data operations
<b>Users</b>	People who interact with the database system

Q2. Define the characteristics of a Database.

Ans: A **database** is more than just a collection of data — it is designed to be **efficient, reliable, and accessible**. Here are the key characteristics that define a good database system:

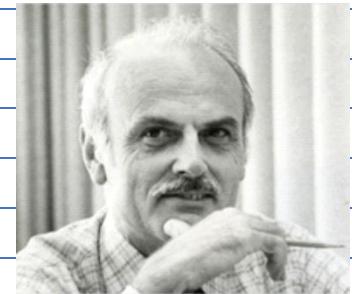
Characteristic	Description
Data Independence	Separates data structure from application logic
Data Integrity	Maintains correctness of data
Data Security	Protects data from unauthorized access
Redundancy Control	Eliminates duplicate data
Data Consistency	Ensures data accuracy across the system
Concurrent Access	Supports multi-user access safely
Backup and Recovery	Restores data after failures
Scalability	Handles increased data and user loads
Query Language Support	Enables easy data manipulation and retrieval

**Q3.** What is Relational Data Model of a database? Explain in detail.

Ans: The **Relational Data Model** is a type of **data model** used to organize data into **tables (called relations)** that are logically connected. It was proposed by **E.F. Codd** in **1970** and is the foundation of most modern **Relational Database Management Systems (RDBMS)** like MySQL, Oracle, SQL Server, and PostgreSQL. The **Relational Data Model** stores data in the form of related tables. It is efficient, flexible, and widely used in real-world applications like banking, education, e-commerce, etc.

### Advantages:

- Easy to understand and use.
- Supports powerful query languages (like SQL).
- Ensures data consistency and integrity.
- Easier to maintain and extend.



Edgar F. Codd  
The Father of the Relational Database

### Example: Students Table

- **Relation:** Students
- **Attributes:** StudentID, Name, Age
- **Tuples:** Each row (101, Alice, 17) is a tuple
- **Primary Key:** StudentID

StudentID	Name	Age
101	Alice	17
102	Bob	18

**Q4.** What is SQL? Explain sub language of SQL.

Ans: **SQL** (Structured Query Language) is a **standard programming language** used to **manage and manipulate relational databases**. It allows users to **create, retrieve, update, and delete** data stored in a relational database. SQL is supported by most RDBMSs (Relational Database Management Systems) such as **MySQL, Oracle, PostgreSQL, SQL Server**, and others.

**Sub-Languages of SQL :** SQL is divided into several **sub-languages**, each designed for specific types of tasks. The main sub-languages of SQL are:

Sub-Language	Purpose	Examples
DDL	Define/alter database structure	CREATE, ALTER, DROP
DML	Manipulate data	INSERT, UPDATE, DELETE
DQL	Query data	SELECT
DCL	Control access	GRANT, REVOKE
TCL	Manage transactions	COMMIT, ROLLBACK, SAVEPOINT

### A: Multiple Choice Questions:

1. Software that plays a supporting role for users and developers, are called \_\_\_\_\_

- A) Operating System      C) Utility Programs
- B) Protection Tools      D) Developer Tools

**Explanation: Utility programs assist users and developers by performing specific system tasks like file management or system optimization.**

2. \_\_\_\_\_ picks up all the scattered pieces of a data across our hard drive and puts them back together again to improve the performance of a computer system.

- A) Fragmentation      C) Disk Cleanup
- B) Defragmentation       D) None of these

**Explanation: Defragmentation reorganizes scattered data on the hard drive to improve system performance.**

3. \_\_\_\_\_ is a system software which acts as an interface between the user and computer hardware.

- A) Utility Program      C) Antivirus Program
- B) Protection System      D) Operating System

**Explanation: An operating system acts as an interface between the user and computer hardware.**

4. Which of the following mode is a diagnostic mode of computer operating system?

- A) Developer Mode      C) Safe Mode
- B) Operating Mode      D) Diagnostic Mode

**Explanation: Safe Mode is a diagnostic mode used to troubleshoot issues by loading only essential system files.**

5. Which of the following utility can be used to protect our system from malwares?

- A) Compression Tools      C) Antivirus Tools
- B) Disk Management Tools      D) None of these

**Explanation: Antivirus tools protect the system from malware and other security threats.**

6. A \_\_\_\_\_ includes bug fixes , and other small improvements.

- A) Software Update       C) Software Upgrade
- B) Disk Management Tool      D) All of these

**Explanation: A software update includes minor changes like bug fixes and small improvements.**

7. A \_\_\_\_\_ is a security system that does not allow unauthorized access on our system while we are connected to a network

- A) Encryption      C) Firewall
- B) Antivirus      D) all of these

**Explanation: A firewall blocks unauthorized access to a computer system over a network.**

B: Write the Full Forms for the Following :

1. PnP – Plug and Play
2. USB – Universal Serial Bus
3. PCB – Printed Circuit Board
4. DVI – Digital Visual Interface
5. OS – Operating System
6. VGA – Video Graphics Array

C: Short Answer Type Questions:

Q1. What do you mean by Computer Maintenance?

Ans **Computer Maintenance** refers to the regular practices and tasks performed to keep a computer system running efficiently, securely, and reliably. It includes both **hardware** and **software** upkeep.

**Common Tasks in Computer Maintenance:**

- Cleaning the computer (physically and digitally).
- Updating operating systems and drivers.
- Defragmenting hard drives (for HDDs).
- Backing up data regularly.
- Removing unnecessary files and programs.
- Scanning for and removing malware.

**Importance:**

- Improves computer performance.
- Extends the life of the system.
- Protects data and privacy.
- Prevents unexpected failures.

Q2 Write any four techniques to improve the performance of computer system?

Ans Here are **four techniques to improve the performance of a computer system**:

1. **Uninstall Unnecessary Programs**
  - Removing unused software frees up system resources like memory and storage, making the computer run faster.
2. **Upgrade RAM (Memory)**
  - Adding more RAM helps the computer handle more tasks at once, improving speed and performance, especially during multitasking.
3. **Use an SSD Instead of HDD**
  - Replacing a traditional hard drive with a Solid State Drive (SSD) can significantly increase boot time, file access speed, and overall system responsiveness.
4. **Run Regular Disk Cleanup and Defragmentation**
  - Cleaning up temporary files and defragmenting (on HDDs) helps organize data more efficiently, speeding up file access and system operations.

**Q3. What is booting?**

**Ans** Booting is the process of starting a computer and loading the operating system into RAM (Random Access Memory) so that it becomes ready for use.

### Types of Booting:

#### 1. Cold Booting (Hard Booting):

- Starting the computer from a completely powered-off state.
- Example: Turning on the computer using the power button.

#### 2. Warm Booting (Soft Booting):

- Restarting the computer without turning off the power.
- Example: Clicking "Restart" in the Start menu.

### Steps in Booting:

1. Power is turned on.
2. BIOS/UEFI runs a hardware check called **POST (Power-On Self Test)**.
3. The system locates and loads the **Operating System** (like Windows or Linux) from the hard drive or SSD.
4. The computer is ready for use.

**Q4. Write the name of different types of Ports available in the modern computers?**

**Ans 1. USB Port (Universal Serial Bus)**

- Used to connect devices like keyboards, mice, flash drives, and printers.

#### 2. HDMI Port (High-Definition Multimedia Interface)

- Used to connect monitors, TVs, and projectors for audio and video output.

#### 3. Ethernet Port (RJ-45)

- Used for wired internet/network connections.

#### 4. Audio Jack (3.5 mm port)

- Used for headphones, microphones, and speakers.

#### 5. VGA Port (Video Graphics Array) (*less common now*)

- Used for older monitors and projectors.

**Q5. How will you add new fonts in the computer system?**

**Ans** Here are the steps to add new fonts to a Windows computer:

### The Control Panel

#### 1. Open Control Panel

- Go to **Start Menu** → search for **Control Panel**.

#### 2. Click on "Fonts"

- In the Control Panel, find and click **Fonts**.

#### 3. Drag and Drop the Font File

- Drag your downloaded font file into the Fonts window.
- The system will automatically install it.

### Q6. What are Utility Programs?

Ans **Utility programs** are special types of software designed to help manage, maintain, and optimize a computer system. They perform specific tasks to support the overall functioning of the computer.

**Utility programs** are tools that help keep your computer running smoothly, safely, and efficiently.

#### Examples of Utility Programs:

1. **Antivirus Software** – Protects the computer from malware (e.g., Windows Defender, Avast).
2. **Disk Cleanup Tool** – Deletes temporary and unnecessary files.
3. **Backup Software** – Helps create copies of important files and restore them if needed.
4. **File Compression Tools** – Reduces file size (e.g., WinRAR, 7-Zip).
5. **Defragmentation Tool** – Organizes data on a hard disk to speed up access (mainly for HDDs).
6. **Firewall** – Monitors network traffic to block unauthorized access.

### Q7. What do you mean by Device Driver?

Ans A **device driver** is a special type of software that allows your computer's operating system to communicate with hardware devices.

#### Examples of Device Drivers:

- **Printer Driver** – Lets the computer send print commands to a printer.
- **Graphics Driver** – Controls how images and videos are displayed.
- **Sound Driver** – Enables audio output through speakers or headphones.
- **Network Driver** – Manages internet or network connections.

### D: Long Answer Type Questions:

**Q1.** What is Hardware Maintenance? Explain some guidelines for maintaining hardware?

**Ans** **Hardware Maintenance** refers to the process of regularly checking, cleaning, repairing, and upgrading the **physical components** of a computer system to ensure it works efficiently and lasts longer. Proper hardware maintenance improves system performance, prevents breakdowns, and increases the life span of your devices.

#### Guidelines for Maintaining Hardware:

##### 1. Keep Hardware Clean

- o Regularly dust off the computer, keyboard, and monitor using a soft cloth or air blower.
- o Avoid eating or drinking near computers.

##### 2. Ensure Proper Ventilation

- o Keep the system in a cool, open space to avoid overheating.
- o Clean air vents and fans regularly.

##### 3. Handle Devices Carefully

- o Avoid dropping or shaking hardware.
- o Use surge protectors to prevent damage from power fluctuations.

##### 4. Use Original Accessories

- o Use genuine cables, chargers, and parts to avoid compatibility or safety issues.

##### 5. Regularly Check for Damage

- o Inspect cables, ports, and connectors for signs of wear or damage.
- o Replace faulty hardware promptly.

##### 6. Update Firmware/Drivers

- o Keep device drivers and firmware up to date for better performance and compatibility.

##### 7. Shut Down Properly

- o Always turn off the computer using the operating system rather than unplugging it directly.

**Q2** What are ports? Explain any two ports in detail?

**Ans** **Hardware ports** are physical interfaces on a computer or other electronic devices that allow you to connect peripherals (external devices) such as keyboards, mice, printers, external storage, monitors, and network cables. These ports serve as communication points where data is transmitted between the computer and the connected device.

There are various types of ports, each designed for specific purposes, with different shapes and data transfer capabilities.

### Two Common Hardware Ports Explained in Detail

#### 1. USB (Universal Serial Bus) Port

##### Function:

USB ports are used to connect a wide range of devices such as keyboards, mice, printers, external hard drives, flash drives, smartphones, and more.

##### Types:

- **USB-A:** The most common rectangular port.
- **USB-B:** Typically used for printers.
- **USB-C:** Newer, reversible, and faster data transfer and charging.
- **Micro-USB / Mini-USB:** Used mainly in older mobile devices.

##### Features:

- Plug and play support.
- Supports hot swapping (devices can be connected/disconnected without restarting).
- Can transmit both data and power.
- Data transfer rates vary (USB 2.0 up to 480 Mbps, USB 3.0 up to 5 Gbps, USB 3.1 and above even faster).

##### Use Case Example:

Charging your phone or transferring files from a flash drive.

#### 2. HDMI (High-Definition Multimedia Interface) Port

##### Function:

HDMI ports are used to transmit high-quality audio and video signals from a device to a display like a TV, monitor, or projector.

##### Features:

- Supports both video and audio in a single cable.
- Capable of transmitting high-definition and even 4K video.
- Comes in standard, mini, and micro sizes.
- Common in TVs, laptops, game consoles, and media players.

##### Use Case Example:

Connecting a laptop to a projector for a presentation or linking a gaming console to a television.

**Q3.** What do you mean by PC Security? Explain any two techniques to secure your PC.

**Ans** **PC Security** refers to the measures and techniques used to protect a personal computer (PC) and its data from unauthorized access, damage, theft, viruses, malware, and other security threats. It involves both hardware and software strategies to ensure the **confidentiality, integrity, and availability** of data and system functionality.  
PC security is essential for protecting sensitive information, maintaining system performance, and preventing cyber-attacks.

### Two Techniques to Secure Your PC

#### 1. Antivirus Software

##### **Explanation:**

Antivirus software is a program designed to detect, prevent, and remove malware (like viruses, worms, spyware, ransomware, etc.) from your PC. It regularly scans your files and system for malicious code and warns or automatically removes threats.

##### **Features:**

- Real-time protection.
- Automatic updates to detect new viruses.
- Quarantine and removal of infected files.

##### **Example:**

Software like **Windows Defender**, **Norton Antivirus**, or **McAfee** protects your PC from threats you might encounter while browsing the internet or opening email attachments.

#### 2. Firewall

##### **Explanation:**

A firewall is a security system that monitors and controls incoming and outgoing network traffic based on predetermined security rules. It acts as a barrier between your PC and potentially harmful external networks (like the internet).

##### **Types:**

- **Software firewall** (runs on your PC)
- **Hardware firewall** (built into routers)

##### **Features:**

- Blocks unauthorized access.
- Prevents hackers from gaining control of your system.
- Can be customized to allow or block specific applications.

##### **Example:**

The built-in **Windows Firewall** helps protect your PC by blocking suspicious activity from the internet.

**Q4. Differentiate between Update and Upgrade.**

Ans Difference Between **Update** and **Upgrade**

<b>Aspect</b>	<b>Update</b>	<b>Upgrade</b>
<b>Definition</b>	A small change or patch to fix bugs, improve security, or enhance features in existing software.	A major change that replaces the existing version with a newer, often significantly different version.
<b>Purpose</b>	To fix issues, patch vulnerabilities, and improve performance.	To add new features, interface changes, or major improvements.
<b>Scope</b>	Minor changes – typically within the same version.	Major changes – involves a new version or generation.
<b>Example</b>	Updating Windows 10 with security patches.	Upgrading from Windows 10 to Windows 11.
<b>Frequency</b>	Happens more frequently (weekly/monthly).	Happens less frequently (yearly or when major release occurs).
<b>Cost</b>	Usually, free.	May be free or paid, depending on the software.

**In simple terms:**

- **Update** = Small fix or improvement.
- **Upgrade** = Big change or new version.

### A: Multiple Choice Questions:

1. Information Technology Act. 2000(India) comes into existence in INDIA on

- A. 17<sup>th</sup> October 2000
- C. 27<sup>th</sup> November 2003
- B. 1<sup>st</sup> January 2006
- D. 17<sup>th</sup> October 2009

**Explanation: The IT Act 2000 was notified on 17th October 2000, making it the day it came into force.**

2. How E-Waste result in data theft ?

- A. By Email Forwarding
- B. **By Replacing Old electronics devices in Exchange without clearing data.**
- C. By Sharing Data
- D. By using Wi-fi network

**Explanation: Data stored in old devices can be recovered and misused if not properly erased before disposal.**

3. Plagiarism means \_\_\_\_\_

- A. **An Act of presenting another person's work or ideas as your own.**
- B. A disease that affect humans and other mammals
- C. A contagious bacterial characterized by fever.
- D. None of Above.

**Explanation: Plagiarism is using someone else's intellectual property (ideas, words, work) without giving credit.**

4. Computer Vandalism \_\_\_\_\_

- A. Action involving deliberate of or damage to property
- B. **Malicious action that involves the destruction of computer and data**
- C. An action to protect computer from virus.
- D. None of Above.

**Explanation: It refers to intentional harm or destruction of computer systems, programs, or data.**

5. On Which toll free number a complaint can be register with Punjab Cyber Crime Division in case of any online fraud?

- A) 1911
- B) **1930**
- C) 1947
- D) 1912

**Explanation: 1930 is the national helpline number in India to report cyber fraud and financial fraud.**

6. Which Act in India Focus on Cyber Crime ?

- A) Banking Regulation Act 1949
- B) **IT Act 2000**
- C) India Penal Code 1860
- D) CrPC 1973

**Explanation: The Information Technology Act 2000 is the primary law in India dealing with cybercrime and electronic commerce.**

### B: Fill in the Blanks:

1. Any offence committed against a person or groups of people using digital gadget is known as Cyber Crime.
2. Cyber Etiquette or Netiquette refers to the code of responsible behaviour on the internet.
3. E-Waste management using landfills pollute soil and groundwater.
4. Cyber Espionage is an act of secretly or stealthily gathering some important or secret information with revealing our internet.
5. CIA triad stands for Confidentiality, Integrity, and Availability.

### C: Write True and False for the following statements:

1. CIA triad stands for Confidentiality, Integrity and Availability.  **True**
2. IT act in India came into force in the year 1995.  **False**
3. E-waste is good for human health.  **False**
4. Cyber Ethics ensure that users understand their responsibility for conducting themselves online.  **True**
5. Hacking is used to expose vulnerabilities in system.  **True**
6. Spying is an act of keeping a secret watch on various activities of opponents for intelligence purpose.  **True**

### D: Short Answer Type Questions:

Q1. **Describe Cyber Crime.**

Ans **Cybercrime** refers to criminal activities that involve the use of computers, digital devices, or networks. These crimes are committed through the internet or other forms of digital communication and can target individuals, organizations, or even governments. Cybercrime can range from simple scams to highly sophisticated attacks that cause major disruption or theft.

**Consequences of Cybercrime:** Financial loss, Loss of privacy and data, Reputation damage  
Emotional distress, National security threats

Q2. **What do you mean by Hacking.**

Ans **Hacking** refers to the act of gaining unauthorized access to computer systems, networks, or digital devices. It often involves bypassing security measures to steal, modify, or destroy data, or to disrupt services.

**Risks of Hacking:**

- Loss of sensitive data
- Financial damage
- Damage to reputation

### Q3. Define with Phishing with example?

Ans **Phishing** is a type of cybercrime where attackers try to trick people into revealing sensitive information—such as usernames, passwords, credit card numbers, or bank account details—by pretending to be a trustworthy source, usually through email, SMS, phone calls, or fake websites.

#### Example of Phishing:

You receive an email that looks like it's from your bank:

**Subject: Urgent: Verify Your Account Now**

Dear Customer,

We have detected unusual activity on your account. For your security, please verify your identity by clicking the link below:

[Click here to verify your account](#)

Failure to act within 24 hours will result in account suspension.

Thank you,

Your Bank's Security Team

If you click the link, you may be taken to a fake website that looks like your bank's login page. If you enter your username and password, the attackers will capture that information and could use it to steal money or commit fraud.

### Q4. Explain CIA Triad

Ans The **CIA Triad** is a fundamental model in cybersecurity that represents the three core principles used to guide policies and practices for securing data and systems. **CIA** stands for: the **CIA Triad—Confidentiality, Integrity, and Availability**—forms the backbone of all cybersecurity efforts and helps ensure the secure handling of information in any organization. The CIA Triad provides a clear framework for thinking about cybersecurity risks and solutions. A breach in **any one** of these areas can lead to serious consequences, such as data leaks, fraud, system outages, or loss of trust.

1. **Confidentiality:** Ensuring that information is accessible only to those authorized to have access.
2. **Integrity:** Ensuring that information is accurate and complete, and has not been altered without authorization.
3. **Availability:** Ensuring that authorized users have reliable access to information and systems when needed.

### Q5. What is Cyber Bullying?

Ans **Cyberbullying** is the use of digital technology—such as social media, messaging apps, websites, or gaming platforms—to harass, threaten, humiliate, or target another person. It is a form of bullying that takes place online and can be just as harmful as face-to-face bullying.

#### Examples:

- A student posts a mean comment about a classmate on Instagram.
- Someone creates a fake profile pretending to be another person to post offensive content.
- A person repeatedly sends insulting messages through a chat app or text.

Cyberbullying is a serious issue in today's digital age. Just like traditional bullying, it can deeply affect a person's mental and emotional well-being. Being respectful and kind online is essential to creating a safe internet environment for everyone.

### Q6. What do you mean by Plagiarism?

Ans **Plagiarism** is the act of using someone else's work, ideas, or words without giving proper credit, and presenting it as your own. It is considered unethical and dishonest, whether done intentionally or unintentionally.

#### Example of Plagiarism:

##### Original Text:

"Cyberbullying is a form of bullying that takes place over digital devices like phones or computers."

##### Plagiarized Version:

Cyberbullying happens when bullying occurs through phones or computers.

*(This is paraphrased but still copied without credit.)*

Plagiarism is a serious offense in both academic and professional settings. Always give credit to the original creators to maintain honesty, respect, and integrity in your work.

### Q7. Explain Identity Theft.

Ans **Identity theft** is a type of cybercrime where someone illegally obtains and uses another person's personal information—such as their name, Social Security number, credit card details, or bank account information—usually to commit fraud or other crimes.

#### Examples of Identity Theft:

- A thief uses your credit card number to make purchases.
- Your social media or email account is taken over and used to scam others.

Identity theft is a serious crime that can damage your finances, reputation, and personal life.

Staying alert, protecting your data, and acting quickly if your information is stolen are key to minimizing harm.

**Q8.** **Describe the Various harmful effects of E-Waste.**

**Ans** **E-waste** (electronic waste) refers to discarded electronic devices like old computers, mobile phones, TVs, batteries, and other gadgets. When not disposed of properly, e-waste poses serious risks to the **environment** and **human health**.

Improper disposal of e-waste has **serious and long-lasting effects** on our environment, health, and economy. Safe recycling, public awareness, and government regulations are essential to reduce the damage caused by e-waste.

### **E:** Long Answer Type Questions:

**Q1.** **What is Cyber Crime? Describe various types of Cyber Crime.**

**Ans** **Cybercrime** refers to any illegal activity that involves the use of computers, digital devices, or the internet. Cybercriminals use technology to access, steal, damage, or manipulate data, systems, or networks for personal, financial, or political gain. Cybercrime is a serious and growing threat in the digital age. It affects individuals, businesses, and governments. Understanding the various types of cybercrime is the first step toward protecting yourself and others in the online world. Strong passwords, antivirus software, and awareness are key to staying safe.

#### **Types of Cybercrime:**

**1. Hacking:** Unauthorized access to a computer or network. Can be used to steal data, disrupt systems, or install malware.

**Example:** Breaking into a company's server to steal customer information.

**2. Phishing:** Fraudulent attempts to trick people into giving personal information by pretending to be a trusted source. Usually done through fake emails or websites.

**Example:** A fake email from a "bank" asking for your login details.

**3. Identity Theft:** Stealing someone's personal information (e.g., name, ID number, credit card details) to commit fraud or crimes in their name.

**Example:** Using someone else's identity to apply for a loan.

**4. Cyberbullying:** Using digital platforms to harass, threaten, or humiliate others. Common on social media, messaging apps, and gaming platforms.

**Example:** Posting insulting or threatening messages online.

**5. Online Fraud and Scams:** Deceiving people to steal money or information. Includes fake shopping websites, lottery scams, and job scams.

**Example:** Paying for a product online that never gets delivered.

**Q2.** Explain Cyber Safety. What are the preventive measures to tackle Cyber Crime?

**Ans** **Cyber Safety** refers to the practices, strategies, and precautions taken to protect yourself, your data, and your devices while using the internet and digital technologies. It involves staying safe from online threats like cybercrime, cyberbullying, identity theft, and data breaches. Cyber safety is essential in today's digital world to protect yourself and your information. By following preventive measures and staying aware of cyber risks, you can greatly reduce your chances of falling victim to cybercrime.

### **Preventive Measures to Tackle Cybercrime:**

#### **1. Use Strong Passwords**

- Create complex passwords with letters, numbers, and symbols.
- Avoid using easily guessable passwords like birthdays or "123456."
- Change passwords regularly.

#### **2. Enable Two-Factor Authentication (2FA)**

- Adds an extra layer of security by requiring a second verification step (like a code sent to your phone).

#### **3. Keep Software Updated**

- Regularly update your operating system, apps, and antivirus software to fix security vulnerabilities.

#### **4. Be Cautious with Emails and Links**

- Don't open emails or attachments from unknown senders.
- Avoid clicking on suspicious links or pop-ups.

#### **5. Use Secure Networks**

- Avoid public Wi-Fi for sensitive transactions.
- Use VPNs (Virtual Private Networks) to encrypt your internet connection.

**Q3. Define Cyber Ethics. What are the principles of Cyber Ethics.**

**Ans** **Cyber Ethics** refers to the moral principles and guidelines that govern the behaviour of individuals when using computers, the internet, and digital technologies. It focuses on using technology responsibly, respecting others' rights, and ensuring fairness and safety online.

### **Principles of Cyber Ethics:**

#### **1. Respect Others' Privacy**

- Do not access or share someone else's personal information without permission.

#### **2. Do Not Harm Others**

- Avoid cyberbullying, harassment, or spreading false information that can hurt people.

#### **3. Respect Intellectual Property**

- Do not steal or copy software, music, videos, or written content without proper authorization or giving credit.

#### **4. Use Technology Fairly and Legally**

- Follow laws related to cybercrime and do not engage in hacking, identity theft, or unauthorized access.

#### **5. Be Honest and Truthful**

- Avoid lying, cheating, or misrepresenting yourself online.

#### **6. Protect Your Own and Others' Security**

- Use strong passwords, update software, and avoid risky online behaviour that can compromise safety.

#### **7. Respect Others' Rights to Freedom of Expression**

- Allow others to share their views, but do so respectfully and without spreading hate speech or offensive content.

#### **8. Practice Accountability**

- Take responsibility for your actions online and understand the consequences.

**Q4.** What do you mean by E-Waste? Explain various methods of disposing E-Waste.

**Ans** **E-Waste** (Electronic Waste) refers to discarded electronic devices or components that are no longer useful or working. This includes items like old computers, mobile phones, televisions, printers, batteries, and other electronic appliances. E-waste contains **toxic materials** such as lead, mercury, and cadmium. If not disposed of properly, it can pollute the environment and harm human health. However, it also contains **valuable materials** like gold, copper, and aluminum that can be recovered through proper recycling.

### Methods of Disposing E-Waste:

#### 1. Recycling

- **Process:** E-waste is collected, dismantled, and useful materials like metals, plastics, and glass are separated and reused.
- **Benefit:** Reduces pollution and saves natural resources.
- **Example:** Recycling old smartphones to extract gold and copper.

#### 2. Reuse and Refurbishing

- **Process:** Old devices are repaired and upgraded for further use.
- **Benefit:** Extends the life of electronic products and reduces waste.
- **Example:** Donating an old but working laptop to a school or charity.

#### 3. Incineration

- **Process:** E-waste is burned at high temperatures to reduce volume.
- **Benefit:** Reduces the amount of waste in landfills.
- **Drawback:** Can release harmful gases if not properly managed.

#### 4. Landfilling (Not Recommended)

- **Process:** Dumping e-waste in landfills.
- **Drawback:** Toxic substances can leach into soil and water, causing serious environmental damage.

#### 5. Take-Back Programs

- **Process:** Many companies offer to take back old electronics for safe recycling or disposal.
- **Example:** Mobile phone brands collecting old phones for recycling.

#### 6. Exchange or Buy-Back Programs

- **Process:** Some manufacturers offer discounts or money in exchange for old devices when buying a new one.
- **Benefit:** Encourages responsible disposal and recycling

## Binary Number System

**(11100101010)<sub>2</sub>**

4 2 1 4 2 1 4 2 1 4 2 1  
(0 1 1 1 0 0 1 0 1 0 1 0)<sub>2</sub>

— — — —

8 4 2 1 8 4 2 1 8 4 2 1  
(0 1 1 1 0 0 1 0 1 0 1 0)<sub>2</sub>

— — —

2048 1024 512 256 128 64 32 16 8 4 2 1  
( 0 1 1 1 0 0 1 0 1 0 1 0 )<sub>2</sub>

## Octal Number System

( — — — — )<sub>8</sub>

## Hexadecimal Number System

( — — — )<sub>16</sub>

## Decimal Number System

( —————— )<sub>10</sub>



## Binary Number System

**(10110101010)<sub>2</sub>**

4 2 1 4 2 1 4 2 1 4 2 1  
(1 0 1 1 0 1 0 1 0 0 1 0)<sub>2</sub>

— — — —

8 4 2 1 8 4 2 1 8 4 2 1  
(1 0 1 1 1 0 1 0 1 0 1 0)<sub>2</sub>

— — —

2048 1024 512 256 128 64 32 16 8 4 2 1  
( 1 0 1 1 1 0 1 0 1 0 1 0 )<sub>2</sub>

## Octal Number System

( — — — — )<sub>8</sub>

## Hexadecimal Number System

( — — — )<sub>16</sub>

## Decimal Number System

( —————— )<sub>10</sub>



**Octal Number System** $(\underline{5} \underline{7} \underline{4} \underline{5})_8$ 
 $(\underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad})_2$   
 4 2 1 4 2 1 4 2 1 4 2 1

8 4 2 1 8 4 2 1 8 4 2 1

 $(\underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad})_2$   
 \_\_\_\_\_

2048 1024 512 256 128 64 32 16 8 4 2 1

 $(\underline{\quad} \underline{\quad} \underline{\quad})_2$   
 \_\_\_\_\_
**Binary Number System** $(\underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad})_2$ **Hexadecimal Number System** $(\underline{\quad} \underline{\quad} \underline{\quad})_{16}$ **Octal Number System** $(\underline{1} \underline{3} \underline{7} \underline{5})_8$ 
 $(\underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad})_2$   
 4 2 1 4 2 1 4 2 1 4 2 1

8 4 2 1 8 4 2 1 8 4 2 1

 $(\underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad})_2$   
 \_\_\_\_\_

2048 1024 512 256 128 64 32 16 8 4 2 1

 $(\underline{\quad} \underline{\quad} \underline{\quad})_2$   
 \_\_\_\_\_
**Binary Number System** $(\underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad} \underline{\quad})_2$ **Hexadecimal Number System** $(\underline{\quad} \underline{\quad} \underline{\quad})_{16}$ 

## Hexadecimal Number System

 $(\text{5 E F})_{16}$ 
 $(\underline{\hspace{1cm}} \quad \underline{\hspace{1cm}} \quad \underline{\hspace{1cm}})_{2}$ 
 $(\underline{\hspace{1cm}} \quad \underline{\hspace{1cm}} \quad \underline{\hspace{1cm}} \quad \underline{\hspace{1cm}})_{2}$ 
 $(\underline{\hspace{1cm}} \quad \underline{\hspace{1cm}} \quad \underline{\hspace{1cm}})_{8}$ 
 $2048 \quad 1024 \quad 512 \quad 256 \quad 128 \quad 64 \quad 32 \quad 16 \quad 8 \quad 4 \quad 2 \quad 1$   
 $(\underline{\hspace{1cm}} \quad \underline{\hspace{1cm}} \quad \underline{\hspace{1cm}})_{2}$ 

Binary Number System

 $(\underline{\hspace{1cm}} \quad \underline{\hspace{1cm}} \quad \underline{\hspace{1cm}} \quad \underline{\hspace{1cm}} \quad \underline{\hspace{1cm}})_{2}$ 

Octal Number System

 $(\underline{\hspace{1cm}} \quad \underline{\hspace{1cm}} \quad \underline{\hspace{1cm}})_{16}$ 

Decimal Number System

 $(\underline{\hspace{1cm}} \quad \underline{\hspace{1cm}} \quad \underline{\hspace{1cm}})_{10}$ 


## Hexadecimal Number System

 $(\text{A C E})_{16}$ 
 $(\underline{\hspace{1cm}} \quad \underline{\hspace{1cm}} \quad \underline{\hspace{1cm}})_{2}$ 
 $(\underline{\hspace{1cm}} \quad \underline{\hspace{1cm}} \quad \underline{\hspace{1cm}} \quad \underline{\hspace{1cm}})_{2}$ 
 $(\underline{\hspace{1cm}} \quad \underline{\hspace{1cm}} \quad \underline{\hspace{1cm}})_{8}$ 
 $2048 \quad 1024 \quad 512 \quad 256 \quad 128 \quad 64 \quad 32 \quad 16 \quad 8 \quad 4 \quad 2 \quad 1$   
 $(\underline{\hspace{1cm}} \quad \underline{\hspace{1cm}} \quad \underline{\hspace{1cm}})_{2}$ 

Binary Number System

 $(\underline{\hspace{1cm}} \quad \underline{\hspace{1cm}} \quad \underline{\hspace{1cm}} \quad \underline{\hspace{1cm}} \quad \underline{\hspace{1cm}})_{2}$ 

Octal Number System

 $(\underline{\hspace{1cm}} \quad \underline{\hspace{1cm}} \quad \underline{\hspace{1cm}})_{16}$ 

Decimal Number System

 $(\underline{\hspace{1cm}} \quad \underline{\hspace{1cm}} \quad \underline{\hspace{1cm}})_{10}$ 

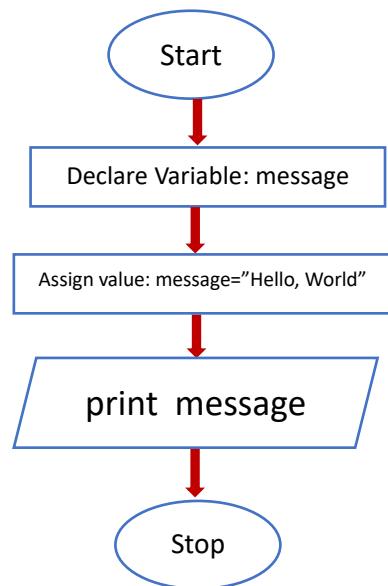

## [Unit II: Python Programming]

**Program 1:** Write a program that prints the message "Hello, World!"

Flow Chart

### Algorithm:

1. **Start**
2. **Define** a variable named message and assign the string value "Hello, World!" to it.
3. **Print** the value of the message variable to the console.
4. **End**



Output of Program:

```
In [1]: runfile('C:/Users/balli/untitled0.py', wdir='C:/Users/balli')
Hello, World!
```

## [Unit II: Python Programming]

**Program 1:** Write a program that prints the message "Hello, World!"

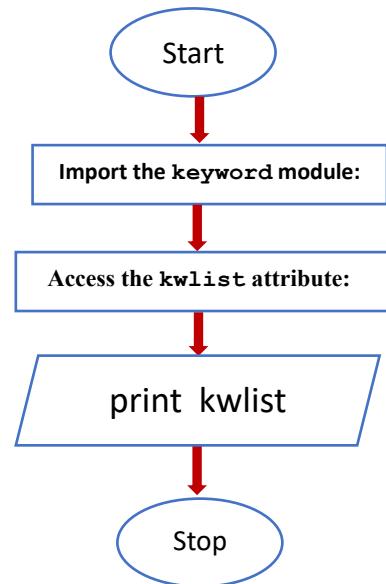
```
def main():
    message="Hello, world"
    print(message)
if __name__ == "__main__":
    main()
```

## Program 2: Write a statement in Python IDLE to view the list of Python keywords.

### Algorithm:

1. Start
2. Import the keyword module:
3. Access the kwlist attribute:
4. Print the kwlist:
5. End

Flow Chart



### Output of Program:

```
In [6]: runfile('C:/Users/balli/untitled0.py', wdir='C:/Users/balli')
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def',
'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda',
'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']
```

```
In [7]:
```

**Program 2:** Write a statement in Python IDLE to view the list of Python keywords.

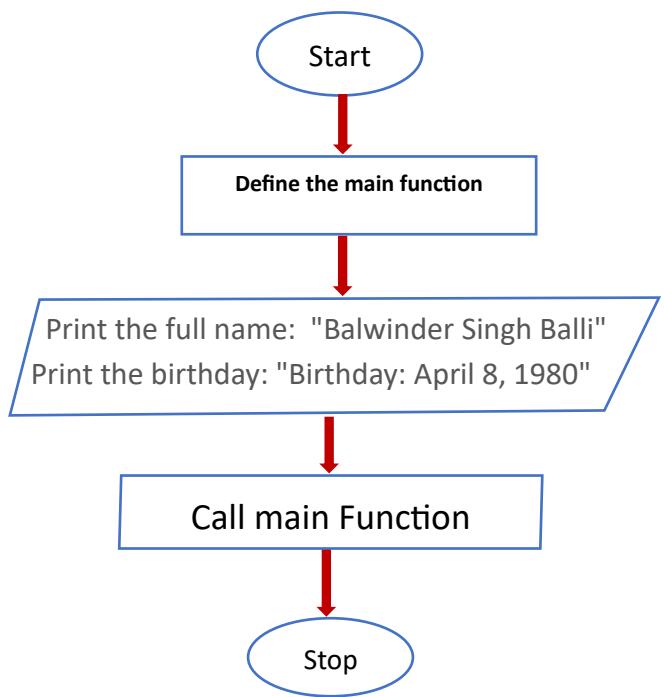
```
import keyword    # Importing the keyword module to access Python's reserved keywords
def main():      # Defining the main function
    print(keyword.kwlist)    # Printing the list of all Python keywords
if __name__ == "__main__": # Checking if this script is being run directly
    main()      # Calling the main function
```

**Program 3:** Write a code that prints your full name and your birthday as separate strings in Python.

Flow Chart

### Algorithm:

1. Start
2. Define a function named main.
3. Inside the main function:  
Print the full name: "Balwinder Singh Balli"  
Print the birthday: "Birthday: April 8, 1980"
4. call the main function.
5. End



### Output of Program:

A screenshot of a Jupyter Notebook interface. The title bar shows 'Console 1/A'. The notebook cell contains the following text:  
In [8]: runfile('D:/balli/cs/11th Class\_2025-26/Program 3.py', wdir='D:/balli/cs/11th Class\_2025-26')  
Balwinder Singh Balli  
Birthday: April 8, 1980

The output of the program is displayed in the cell, showing the printed full name and birthday.

**Program3:** Write a code that prints your full name and your birthday as separate strings in Python.

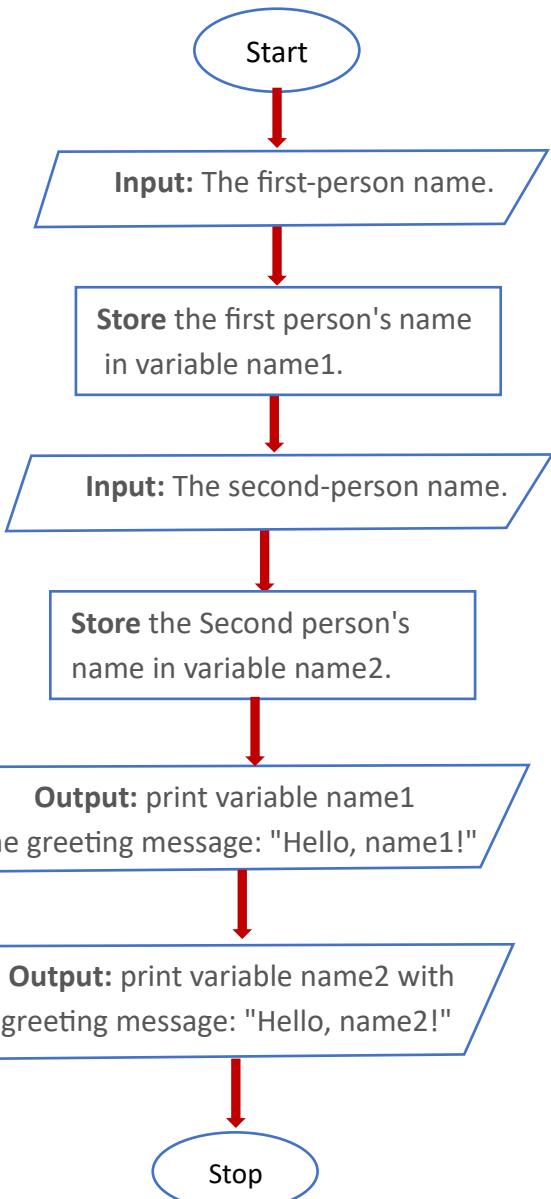
```
def main():                      # Defining the main function
    print("Balwinder Singh Balli")  # name as Separate string
    print("Birthday: April 8, 1980") # Birthday as Separate String
if __name__ == "__main__":
    main()                         # Calling the main function
```

## [Unit II: Python Programming]

**Program 4:** Write a program that asks two people for their names, stores the names in variables name1 and name2, and says hello to both of them.

### Algorithm: Say 'hello' to greet two people

1. Start
2. **Input:** The first-person name.
3. **Store** the first person's name in variable name1.
4. **Input:** The second-person name.
5. **Store** the second person's name in variable name2.
6. **Output:** print Variable name1 with the greeting message: "Hello, name1!"
7. **Output:** print Variable name2 with the greeting message: "Hello, name2!"
8. End



Output of Program :

```
In [13]: runfile('D:/balli/cs/11th Class_2025-26/Program 4.py', wdir='D:/balli/cs/11th Class_2025-26')
Enter the name of the first person: Balwinder Singh balli
Enter the name of the second person: Preetinder Singh protocol
Hello, Balwinder Singh balli!
Hello, Preetinder Singh protocol!
```

```
In [14]:
```

**Program 4:** Write a program that asks two people for their names, stores the names in variables name1 and name2, and says hello to both of them.

```
def main(): # Defining the main function
    # name1 variable declaration
    # Asking for the first person's name
    name1 = input("Enter the name of the first person: ")
    # name2 variable declaration
    # Asking for the second person's name
    name2 = input("Enter the name of the second person: ")
    # Saying hello to both people
    print("Hello, " + name1 + "!")
    print("Hello, " + name2 + "!")

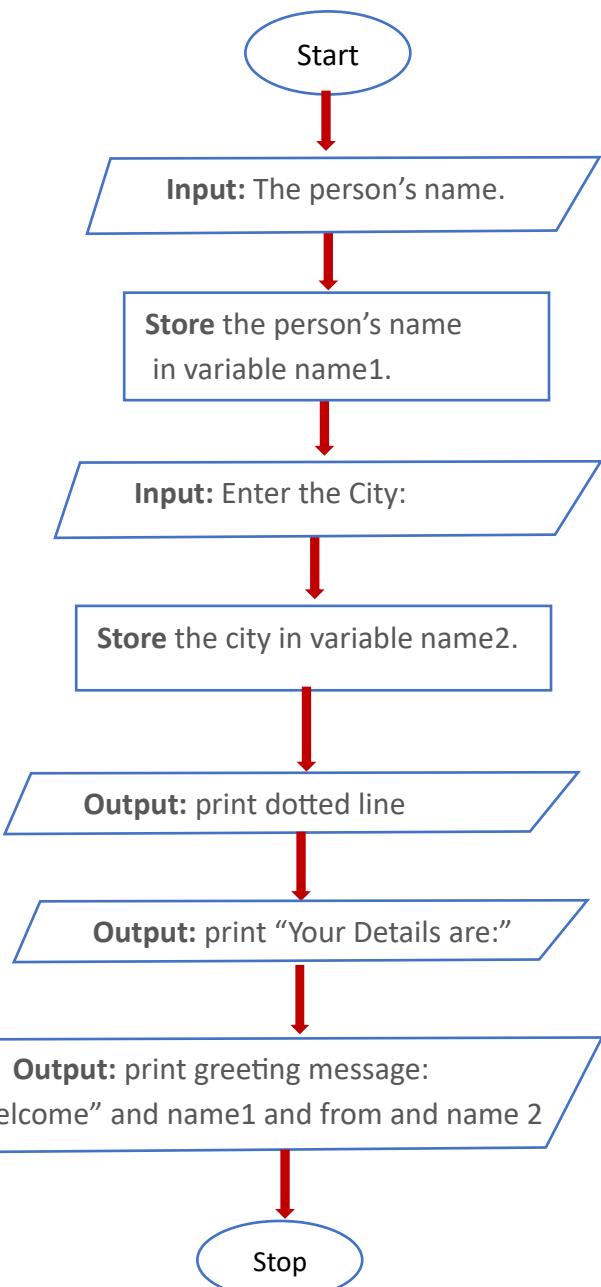
if __name__ == "__main__": # Checking if this script is being run directly
    main() # Calling the main function
```

## [Unit II: Python Programming]

**Program 5:** Write a Python program prompts the user for their name and city, then displays a personalized welcome message.

### Algorithm:

1. **Start:** Begin the program.
2. **Prompt for Name:** Ask the user to input their name and store the input in the name1 variable.
3. **Prompt for City:** Ask the user to input their city and store the input in the name2 variable.
4. **Print Dotted Line:** Print a series of dashed lines for separation using the print() function.
5. **Print "Your details are:"**: Print the message "Your details are:" to introduce the user's information.
6. **Display Personalized Welcome Message:** Concatenate the name1 and name2 variables into a welcome message and print it in the format: "Welcome [name1] from [name2]".
7. **End:** End the program.



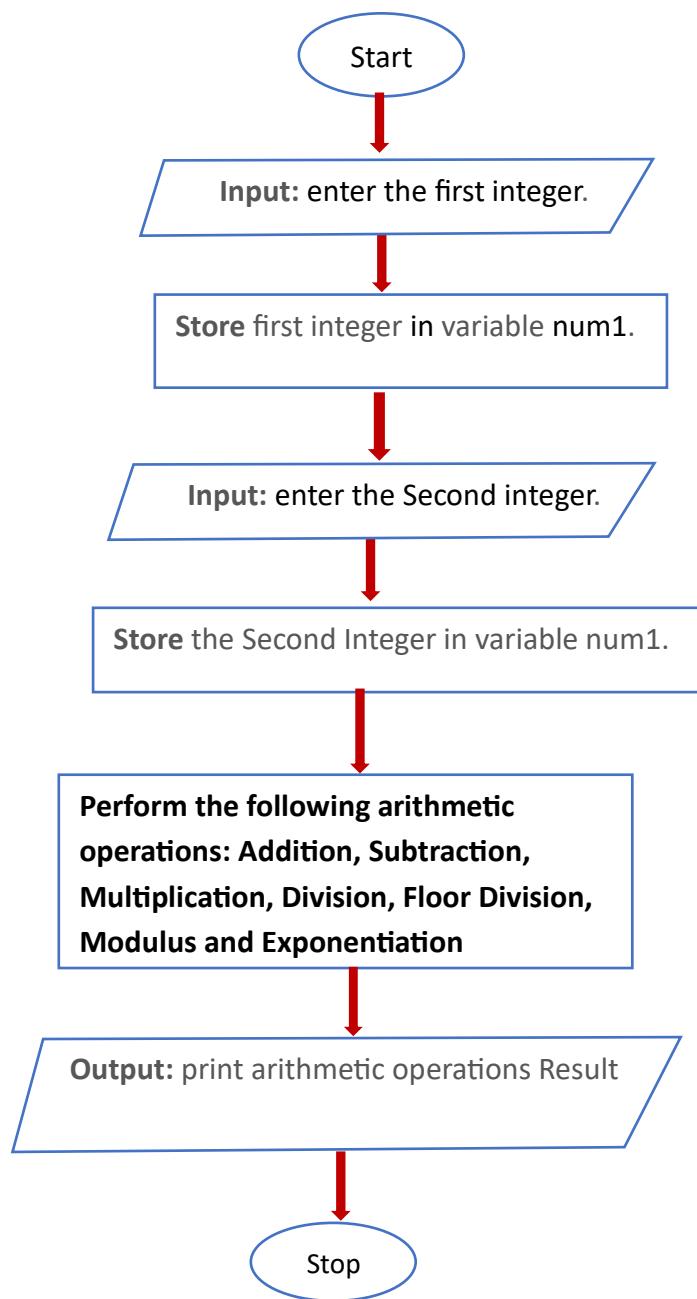
```
In [18]: runfile('D:/balli/cs/11th Class_2025-26/Program 5.py', wdir='D:/balli/cs/11th Class_2025-26')
Enter Your Name:Balwinder Singh Balli
Enter Your City: Patiala
-----
Your details are :
Welcome Balwinder Singh Balli from Patiala
```

**Program 5:** Write a Python program prompts the user for their name and city, then displays a personalized welcome message.

```
def main(): # Defining the main function
    # name1 variable declaration
    name1 = input("Enter Your Name:") # Asking for the person's name
    # name2 variable declaration
    name2 = input("Enter Your City: ") # Asking for the person's City
    print("-----") # add doted line to console
    print("Your details are:")
    print("Welcome " + name1 + " from " + name2) # print final Message with detail
    if __name__ == "__main__": # Checking if this script is being run directly
        main() # Calling the main function
```

### Algorithm: Perform Arithmetic Operations on Two Integers

1. **Start**
2. **Prompt the user** to enter the first integer  
→ Store it in num1
3. **Prompt the user** to enter the second integer → Store it in num2
4. **Perform the following arithmetic operations and display the results:**
  - Add num1 and num2, display the result ("Addition")
  - Subtract num2 from num1, display the result ("Subtraction")
  - Multiply num1 and num2, display the result ("Multiplication")
  - Divide num1 by num2, display the result ("Division")
  - Perform floor division (num1 // num2), display the result ("Floor Division")
  - Find modulus (num1 % num2), display the result ("Modulus")
  - Perform exponentiation (num1 \*\* num2), display the result ("Exponentiation")
5. **End**



```
=====
RESTART: D:/balli/cs/11th Class_2025-26/Chapter3_Python/program1.py =====
Enter first integer: 23
Enter second integer: 3
Addition: 26
Subtraction: 20
Multiplication: 69
Division: 7.666666666666667
Floor Division: 7
Modulus: 2
Exponentiation: 12167
```

**Program 1:** Write a program to enter two integers and perform all Arithmetic Operations on them.

```
def main():
    # Input from the user
    num1 = int(input("Enter first integer: "))
    num2 = int(input("Enter second integer: "))

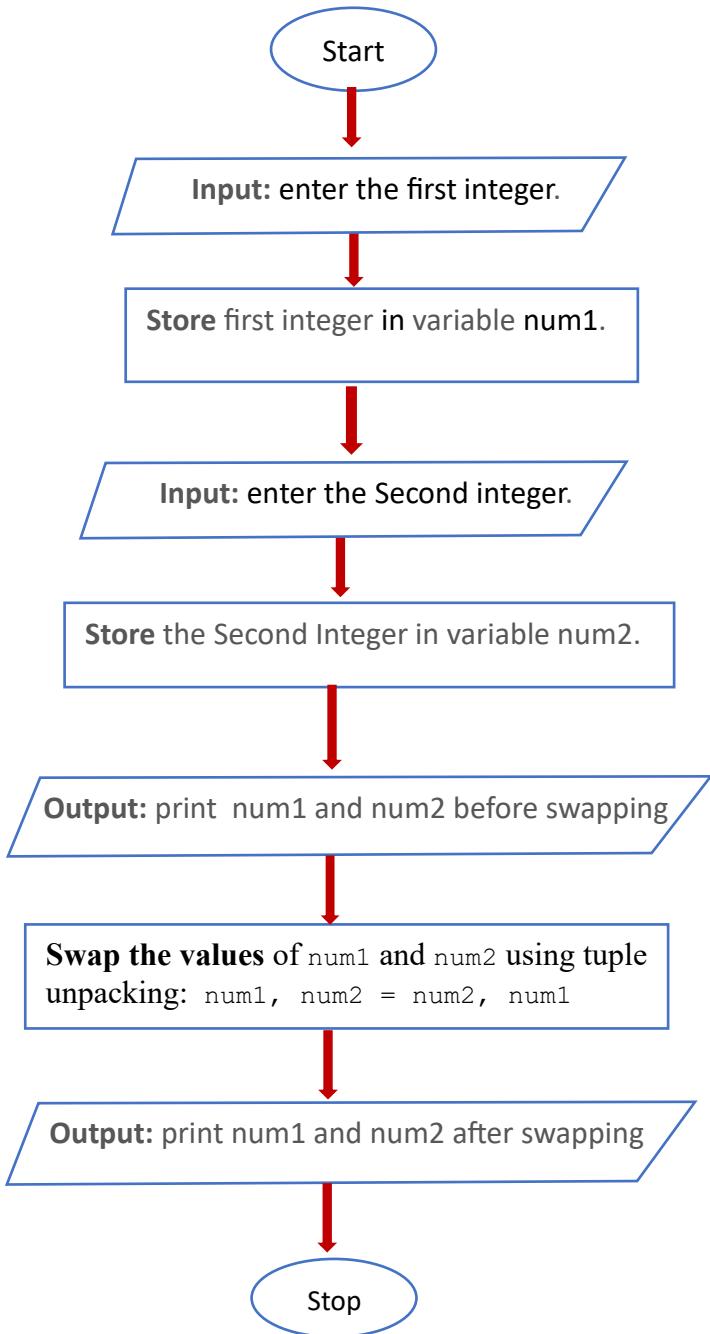
    # Performing arithmetic operations
    print("Addition:", num1 + num2)
    print("Subtraction:", num1 - num2)
    print("Multiplication:", num1 * num2)
    print("Division:", num1 / num2)
    print("Floor Division:", num1 // num2)
    print("Modulus:", num1 % num2)
    print("Exponentiation:", num1 ** num2)

# Call the main function
if __name__ == "__main__":
    main()
```

**Program 2:** Write a Python program to Swap two numbers.

### Algorithm: Swap Two Numbers Without Using a Temporary Variable

1. Start
2. **Prompt the user** to enter the first number  
→ Store the value in variable num1
3. **Prompt the user** to enter the second number  
→ Store the value in variable num2
4. **Display the values** of num1 and num2 before swapping
5. **Swap the values** of num1 and num2 using tuple unpacking:  
`num1, num2 = num2, num1`
6. **Display the values** of num1 and num2 after swapping
7. End



Enter first number: 23  
Enter second number: 45

Before Swapping:  
num1 = 23  
num2 = 45

After Swapping:  
num1 = 45  
num2 = 23

**Program 2:** Write a Python program to Swap two numbers.

```
def main():
    num1 = int(input("Enter first number: "))
    num2 = int(input("Enter second number: "))

    print("\nBefore Swapping: ")
    print("num1 =", num1)
    print("num2 =", num2)

    # Swapping without temporary variable
    num1, num2 = num2, num1

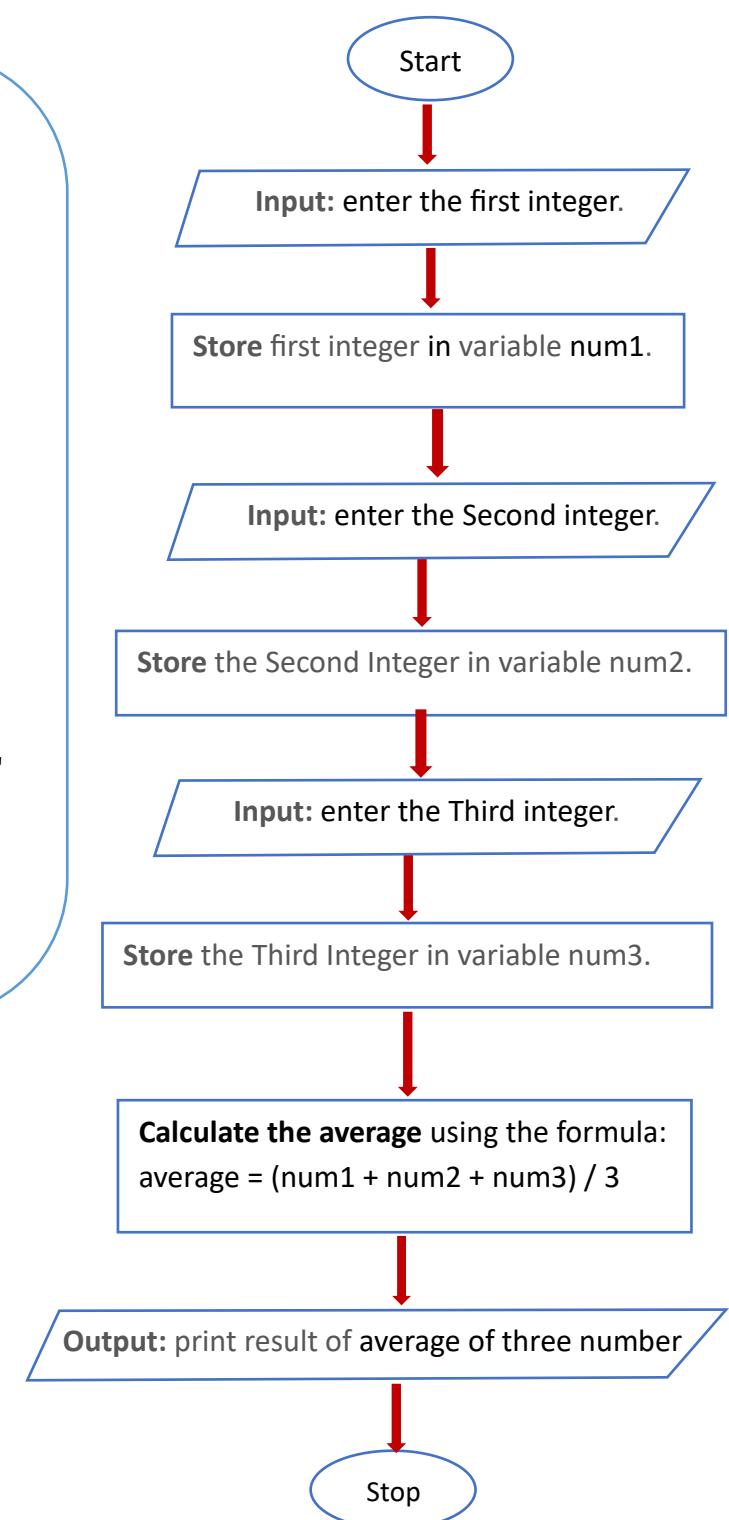
    print("\nAfter Swapping: ")
    print("num1 =", num1)
    print("num2 =", num2)

if __name__ == "__main__":
    main()
```

**Program 3:** Write a Python program that asks three values to find average of three numbers.

### Algorithm: Find the Average of Three Numbers

1. **Start**
2. **Prompt the user** to enter the first number  
→ Store the value in num1
3. **Prompt the user** to enter the second number  
→ Store the value in num2
4. **Prompt the user** to enter the third number  
→ Store the value in num3
5. **Calculate the average** using the formula:  
$$\text{average} = (\text{num1} + \text{num2} + \text{num3}) / 3$$
6. **Display** the result:  
"The average of the three numbers is: average"
7. **End**



Enter first number: 34.5

Enter second number: 45.7

Enter third number: 23.7

The average of the three numbers is: 34.63333333333333

**Program 3:** Write a Python program that asks three values to find average of three numbers.

```
def main():
    # Input three numbers from the user
    num1 = float(input( "Enter first number: " ))
    num2 = float(input( "Enter second number: " ))
    num3 = float(input( "Enter third number: " ))

    # Calculate average
    average = (num1 + num2 + num3) / 3

    # Display the result
    print( "The average of the three numbers is:" , average)

# Call the main function
if __name__ == "__main__":
    main()
```

◆ **Algorithm: Find the Area of a Circle**

1. **Start**
2. **Prompt the user** to enter the radius of the circle  
→ Store the value in variable r
3. **Calculate the area** of the circle using the formula:  
 $\text{area} = \pi \times r \times r$   
(Use  $\pi \approx 3.14159$  or use the math.pi constant)
4. **Display the result**:  
"The area of the circle is: area"
5. **End**

Start

Import the Math package

Input: enter the radius.

Store the value in variable2

Calculate the area of the circle using the formula:  $\text{area} = \pi \times r \times r$ 

Output: print The area of the circle is: area

Stop

```
Enter the radius of the circle: 2
The area of the circle is: 12.566370614359172
```

```
Enter the radius of the circle: 2.56
The area of the circle is: 20.58874161456607
```

**Program 4:** Write a Program that asks radius (r) to find the area of circle.

```
import math
def main():
    # Input: Ask user to enter the radius
    r = float(input("Enter the radius of the circle: "))

    # Process: Calculate area using the formula: πr2
    area = math.pi * r * r

    # Output: Display the area
    print("The area of the circle is:", area)

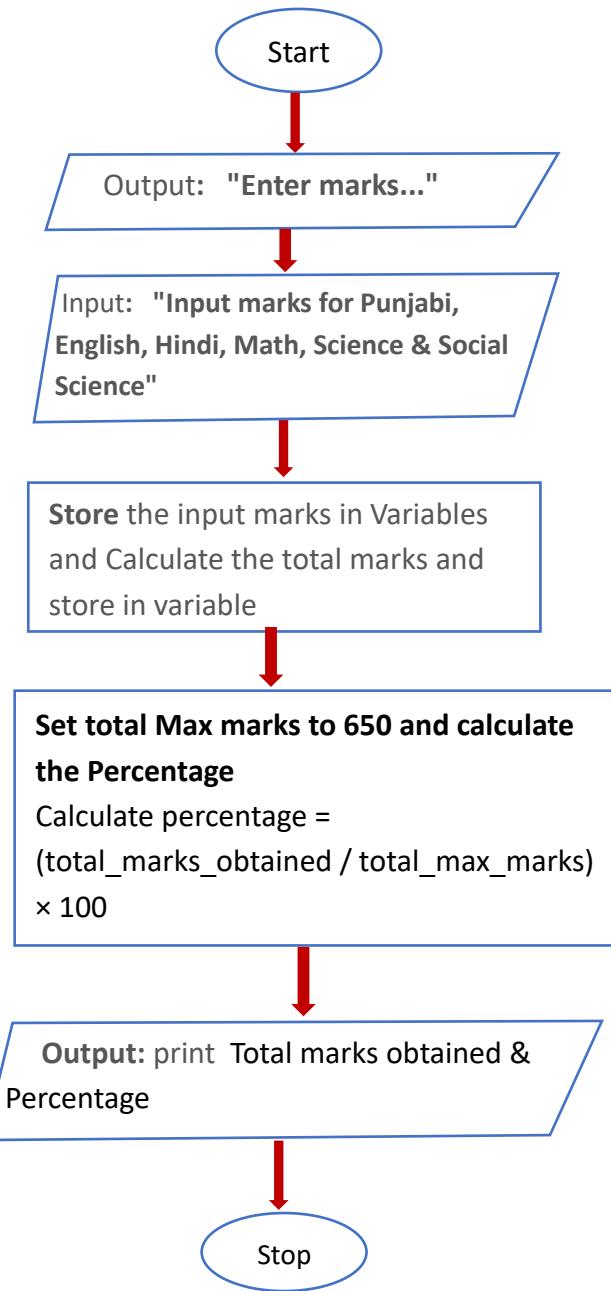
# Call the main function
if __name__ == "__main__":
    main()
```

## [Unit II: Python Programming]

**Program 5:** Write a Program that asks the individual marks of subjects to calculate the percentage of marks of your 10<sup>th</sup> class result.

**Algorithm**

1. **Start**
2. Display "Enter the marks obtained in each subject:"
3. Input marks for:
  - o Punjabi (out of 150)
  - o English (out of 100)
  - o Hindi (out of 100)
  - o Math (out of 100)
  - o Science (out of 100)
  - o Social Science (out of 100)
4. Calculate total\_marks\_obtained = sum of all marks
5. Set total\_max\_marks = 650
6. Calculate percentage =
 
$$\left( \frac{\text{total\_marks\_obtained}}{\text{total\_max\_marks}} \times 100 \right)$$
7. Display:
  - o Total marks obtained
  - o Percentage
8. **End**



```

Enter the marks obtained in each subject:
Punjabi (out of 150): 105
English (out of 100): 98
Hindi (out of 100): 78
Math (out of 100): 67
Science (out of 100): 89
Social Science (out of 100): 78

--- Result ---
Total Marks Obtained: 515.0 / 650
Percentage: 79.23%
  
```

**Program 5:** Write a Program that asks the individual marks of subjects to calculate the percentage of marks of your 10<sup>th</sup> class result.

```
def main():
    print("Enter the marks obtained in each subject:")
    # Input marks for each subject
    punjabi = float(input("Punjabi (out of 150): "))
    english = float(input("English (out of 100): "))
    hindi = float(input("Hindi (out of 100): "))
    math = float(input("Math (out of 100): "))
    science = float(input("Science (out of 100): "))
    social_science = float(input("Social Science (out of 100): "))

    # Total and percentage calculation
    total_marks_obtained = punjabi + english + hindi + math + science + social_science
    total_max_marks = 650
    percentage = (total_marks_obtained / total_max_marks) * 100

    # Display result
    print("\n--- Result ---")
    print(f"Total Marks Obtained: {total_marks_obtained} / {total_max_marks}")
    print(f"Percentage: {percentage:.2f}%")

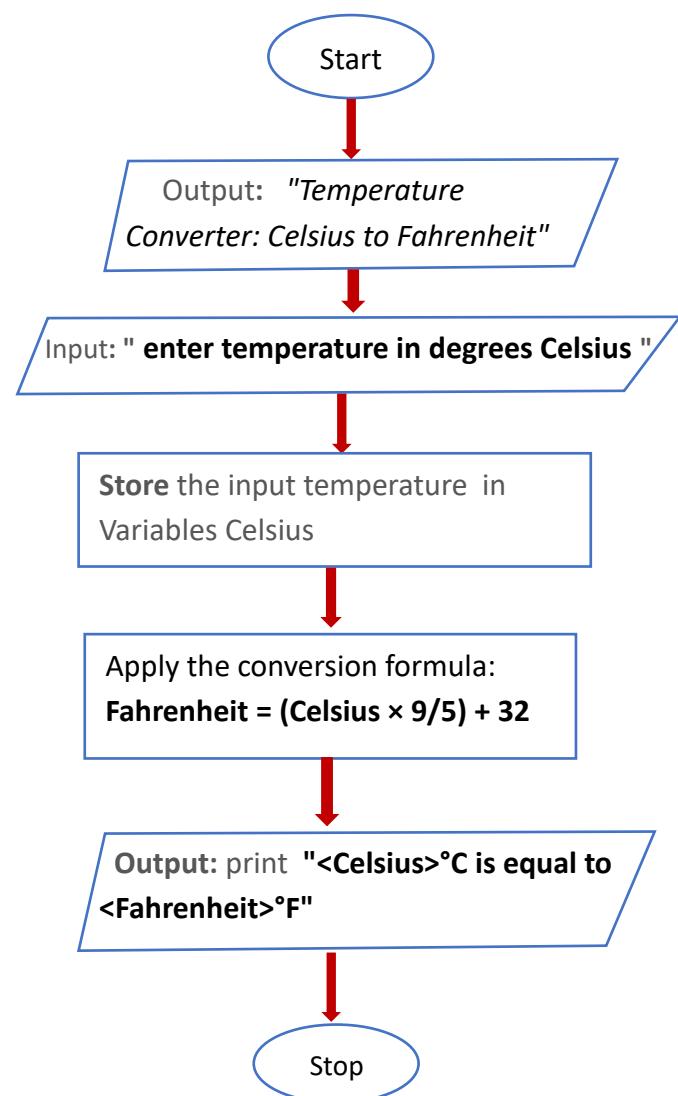
# Call the main function
if __name__ == "__main__":
    main()
```

**Program 6:** Write a program that asks for the temperature in degrees Celsius and converts it into Fahrenheit using the formula:  $F = C \times 9/5 + 32$ .

### Algorithm: Convert Celsius to Fahrenheit

1. Start
2. Display the message: "Temperature Converter: Celsius to Fahrenheit"
3. Prompt the user to **enter temperature in degrees Celsius**
4. Read and store the input in variable Celsius
5. Apply the conversion formula:  

$$\text{Fahrenheit} = (\text{Celsius} \times 9/5) + 32$$
6. Display the result:  
 "<Celsius>°C is equal to <Fahrenheit>°F"
7. End



```
===== RESTART: D:/balli/cs/11th Class _2025-26/Chapter3_Python/program6.py =====
Temperature Converter: Celsius to Fahrenheit
Enter temperature in degrees Celsius: 27
27.0°C is equal to 80.60°F
```

**Program 6:** Write a program that asks for the temperature in degrees Celsius and converts it into Fahrenheit using the formula:  $F = C \times 9/5 + 32$ .

```
def main():
    print("Temperature Converter: Celsius to Fahrenheit")

    # Ask the user to enter temperature in degrees Celsius
    celsius = float(input("Enter temperature in degrees Celsius: "))

    # Convert to Fahrenheit using the formula
    fahrenheit = (celsius * 9/5) + 32

    # Display the result
    print(f"{celsius}°C is equal to {fahrenheit:.2f}°F")

# Run the main function
if __name__ == "__main__":
    main()
```

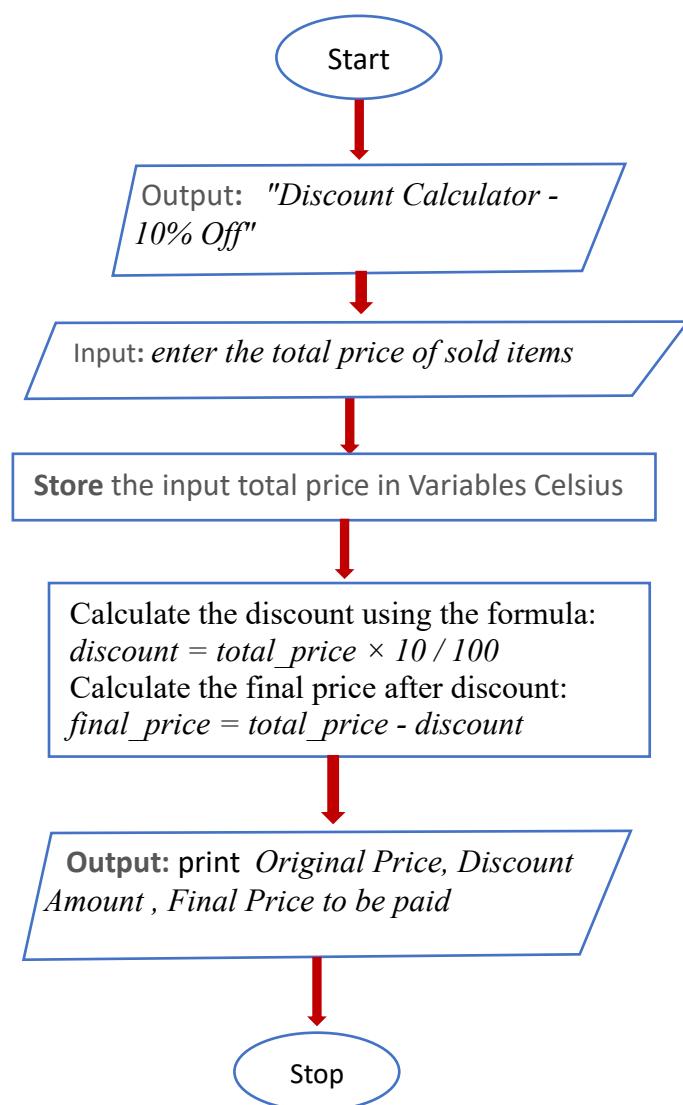
**Program 7:** Write a program to calculate 10% discount on the total price of sold items.

### Algorithm: Calculate 10% Discount on Total Price

1. **Start**
2. Display the message: *"Discount Calculator - 10% Off"*
3. Prompt the user to **enter the total price of sold items**
4. Read and store the input in a variable called `total_price`
5. Calculate the discount using the formula:  

$$\text{discount} = \text{total\_price} \times 10 / 100$$
6. Calculate the final price after discount:  

$$\text{final\_price} = \text{total\_price} - \text{discount}$$
7. Display:
  - o Original Price
  - o Discount Amount
  - o Final Price to be paid
8. **End**



```

=====
RESTART: D:/balli/cs/11th Class_2025-26/Chapter3_Python/program7.py ====
Discount Calculator - 10% Off
Enter the total price of sold items: 300

Original Price: ₹300.00
Discount (10%): ₹30.00
Final Price to be paid: ₹270.00
  
```

## [Unit II: Python Programming]

Date
Page No.

**Program 7:** Write a program to calculate 10% discount on the total price of sold items.

```
def main():
    print( "Discount Calculator - 10% Off" )

    # Ask the user to enter the total price of sold items
    total_price = float(input( "Enter the total price of sold items: " ))

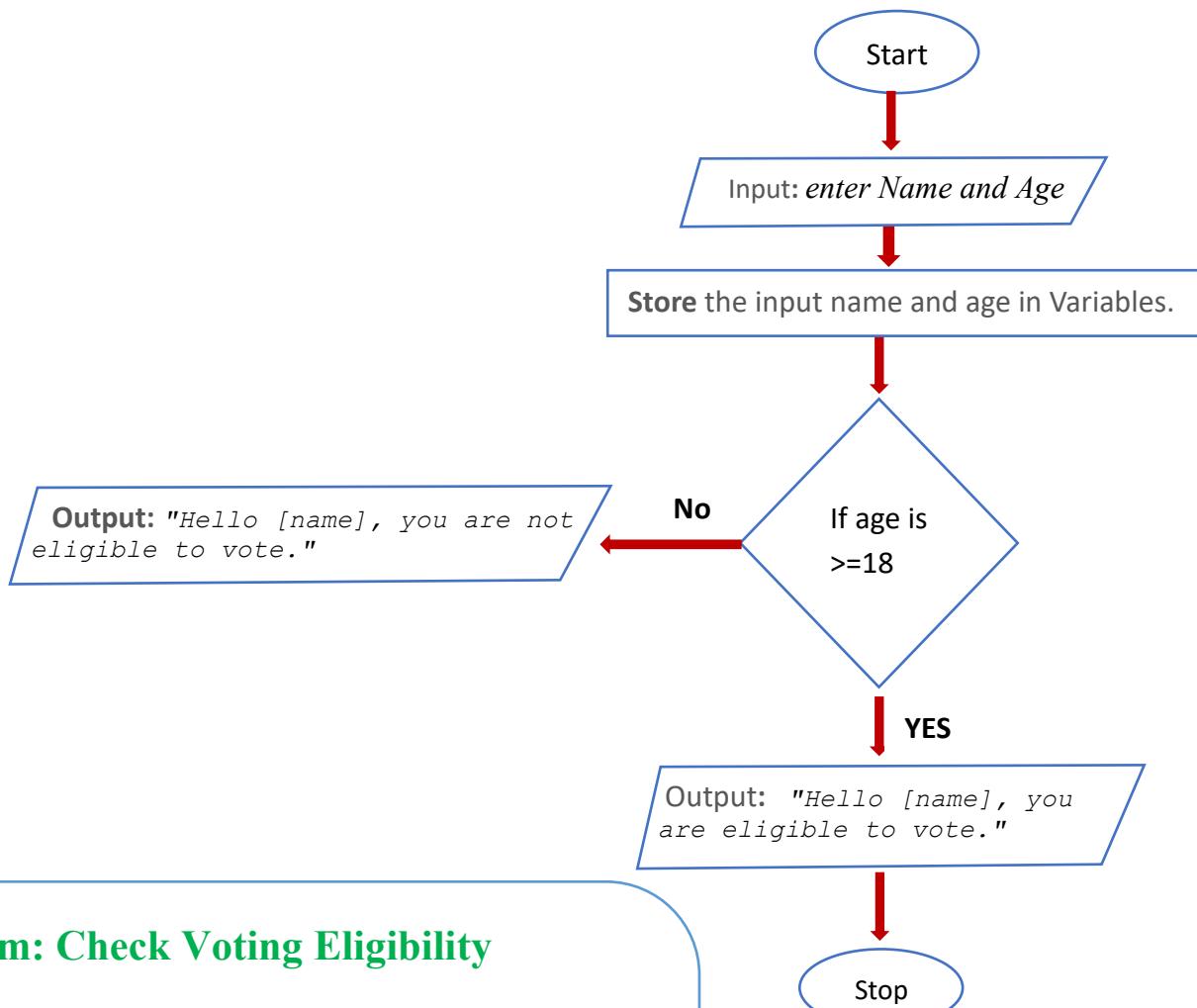
    # Calculate 10% discount
    discount = total_price * 10 / 100

    # Calculate final price after discount
    final_price = total_price - discount

    # Display results
    print( f"\nOriginal Price: ₹{total_price:.2f}" )
    print( f"Discount (10%): ₹{discount:.2f}" )
    print( f"Final Price to be paid: ₹{final_price:.2f}" )

# Run the main function
if __name__ == "__main__":
    main()
```

**Program 1:** Write a program that takes the name and age of the user as input and displays a message whether the user is eligible to vote. ( The eligible age is 18 years ).



### Algorithm: Check Voting Eligibility

1. **Start**
2. Prompt the user to **enter their name** and store it in the variable `name`.
3. Prompt the user to **enter their age** and store it in the variable `age` (convert it to integer)
4. **Check if age is greater than or equal to 18**
  - **If True**, display: "Hello [name], you are eligible to vote."
  - **Else**, display: "Hello [name], you are not eligible to vote."
5. **End**

```
===== RESTART: D:/balli/cs/11th Class_2025-26/Chapter 4_Python/Program2.py =====
Enter your name: Balwinder Singh balli
Enter your age: 45
Hello Balwinder Singh balli, you are eligible to vote.
```

```
===== RESTART: D:/balli/cs/11th Class_2025-26/Chapter 4_Python/Program2.py =====
Enter your name: Preetinder Singh
Enter your age: 17
Hello Preetinder Singh , you are not eligible to vote.
```

**Program 1:** Write a program that takes the name and age of the user as input and displays a message whether the user is eligible to vote. ( The eligible age is 18 years ).

```
def main():
    # Take user's name and age as input
    name = input( "Enter your name: " )
    age = int(input( "Enter your age: " ))

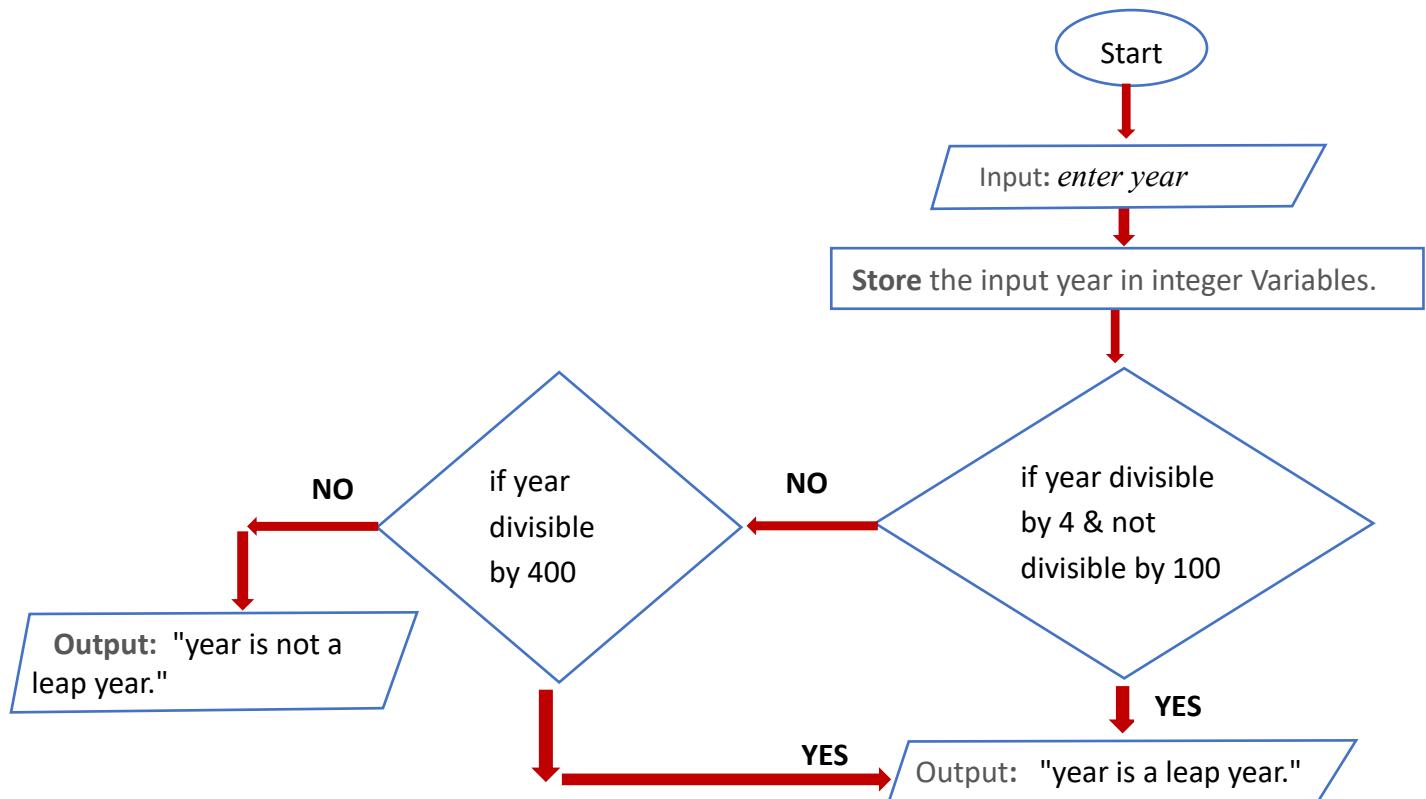
    # Check voting eligibility
    if age >= 18:
        print( f"Hello {name}, you are eligible to vote." )
    else:
        print( f"Hello {name}, you are not eligible to vote." )

# Call the main function
if __name__ == "__main__":
    main()
```

## Chapter-4 Control Statements

### Unit -II Python programming

Program 2: Write a program to check if the year entered by the user is a leap year or not.



#### Algorithm to Check if a Year is a Leap Year

1. **Start**
2. **Prompt** the user to enter a year.
3. **Read** the input and convert it to an integer, store it in the variable year.
4. **Check** if the year is divisible by 4 **and** not divisible by 100:
  - o If **true**, go to step 6.
  - o If **false**, go to step 5.
5. **Check** if the year is divisible by 400:
  - o If **true**, go to step 6.
  - o If **false**, go to step 7.
6. **Print** "year is a leap year."
7. **Print** "year is not a leap year."
8. **End**

```
===== RESTART: D:/balli/cs/11th Class_2025-26/Chapter 4_Python/Program3.py =====
Enter a year: 2025
2025 is not a leap year.
```

```
===== RESTART: D:/balli/cs/11th Class_2025-26/Chapter 4_Python/Program3.py =====
Enter a year: 2024
2024 is a leap year.
```

**Program 2:** Write a program to check if the year entered by the user is a leap year or not.

```
def main():
    # Get input from the user
    year = int(input("Enter a year: "))
    # Check if it is a leap year
    if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
        print( f"{year} is a leap year." )
    else:
        print( f"{year} is not a leap year." )
# Call the main function
if __name__ == "__main__":
    main()
```

## Chapter-4 Control Statements

### Unit -II Python programming

**Program 3:** Write a program to input the percentage of marks of a student and then find the grade of student as per the conditions given below.

#### Percentage of Marks

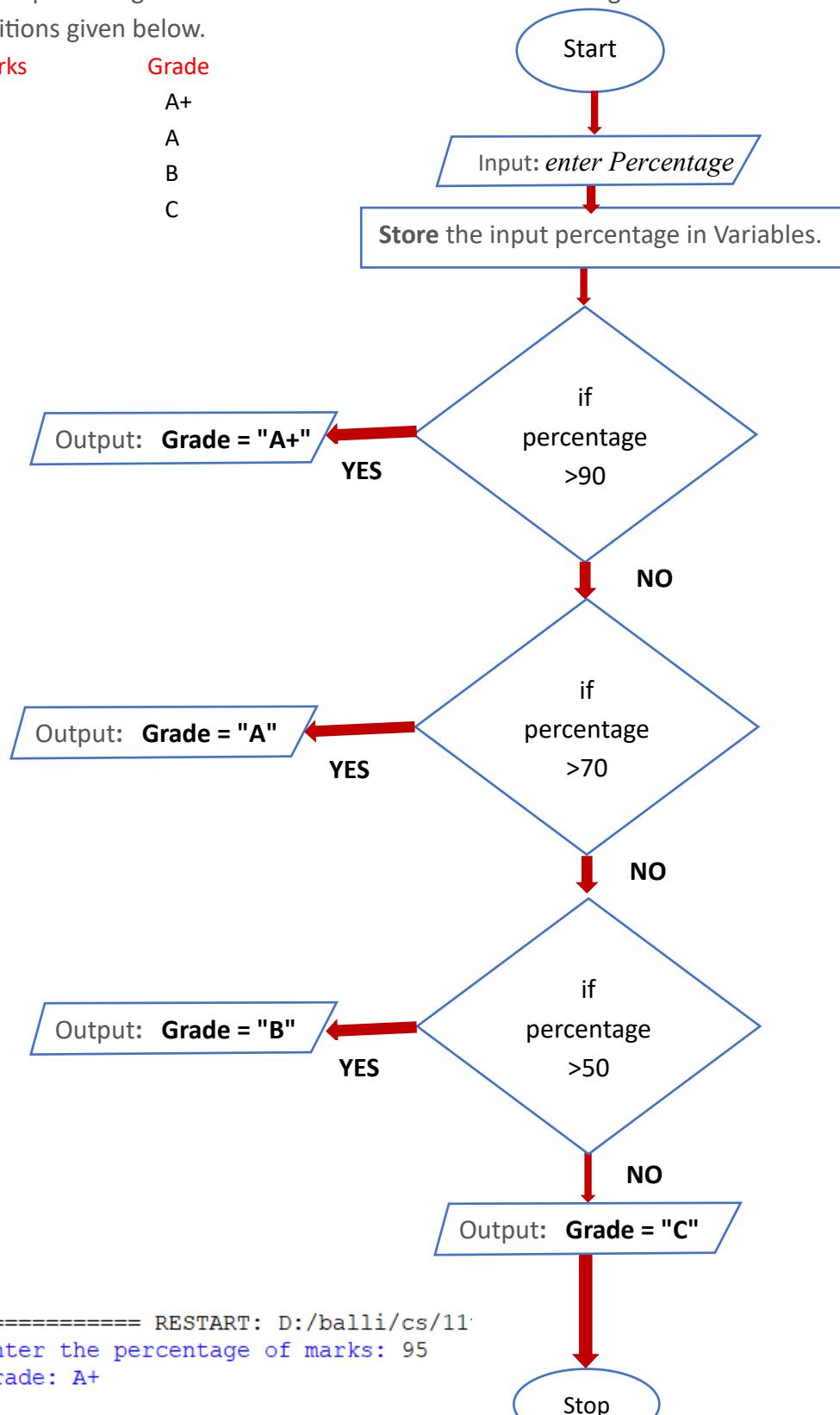
Above 90  
70 to 90  
50 to 70  
Below 50

#### Grade

A+  
A  
B  
C

### Algorithm to Determine Student Grade Based on Percentage

1. Start
2. Prompt the user to enter the percentage of marks.
3. Read and store the percentage in a variable percentage.
4. Check if percentage > 90
  - o If true, assign grade = "A+" and go to step 8.
5. Check if percentage  $\geq 70$ 
  - o If true, assign grade = "A" and go to step 8.
6. Check if percentage  $\geq 50$ 
  - o If true, assign grade = "B" and go to step 8.
7. If none of the above, assign grade = "C"
8. Display the grade.
9. End



```
===== RESTART: D:/balli/cs/11
Enter the percentage of marks: 95
Grade: A+
```

```
===== RESTART: D:/balli/cs/11
Enter the percentage of marks: 65
Grade: B
```

```
===== RESTART: D:/balli/cs/11
Enter the percentage of marks: 75
Grade: A
```

```
===== RESTART: D:/balli/cs/11
Enter the percentage of marks: 45
Grade: C
```

**Program 3:** Write a program to input the percentage of marks of a student and then find the grade of student as per the conditions given below.

Percentage of Marks	Grade
Above 90	A+
70 to 90	A
50 to 70	B
Below 50	C

```
def main():
    percentage = float(input("Enter the percentage of marks: "))
    # Determine the grade
    if percentage > 90:
        grade = "A+"
    elif percentage >= 70:
        grade = "A"
    elif percentage >= 50:
        grade = "B"
    else:
        grade = "C"
    # Display the result
    print(f"Grade: {grade}")
# Call the main function
if __name__ == "__main__":
    main()
```

## Chapter-4 Control Statements

**Program 4:** Write a program to create a simple calculator to perform basic arithmetic operations on two numbers. The program should do the following:

- Accept two numbers from the user.
- Ask user to input any of the operator(+,-,\*,/). An error message is displayed if the user enters anything else.
- As per the input of operator, do the corresponding operation on numbers entered by user.

### Algorithm: Simple Calculator

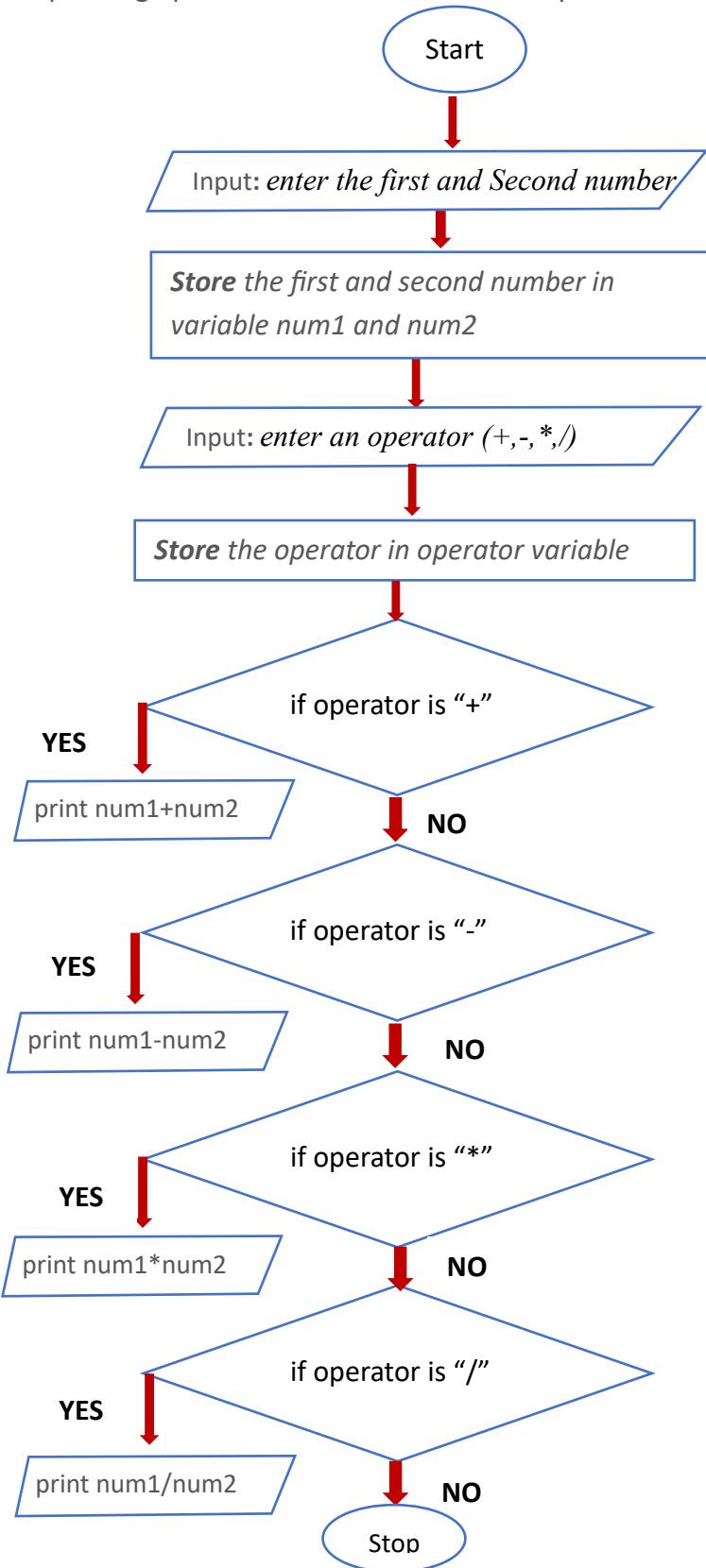
- Start**
- Prompt** the user to enter the first number.
- Store** the input as num1 after converting it to a float.
- Prompt** the user to enter the second number.
- Store** the input as num2 after converting it to a float.
- Prompt** the user to enter an operator (+, -, \*, or /).
- Store** the input as operator.
- Check** the value of operator:
  - If operator is '+', then:
    - Calculate result = num1 + num2
    - Display** the result.
  - Else if operator is '-', then:
    - Calculate result = num1 - num2
    - Display** the result.
  - Else if operator is '\*', then:
    - Calculate result = num1 \* num2
    - Display** the result.
  - Else if operator is '/', then:
    - calculate result = num1 / num2
    - display** the result.
- End**

```
===== RESTART: D:/balli/cs/11th Class_2
Enter the first number: 23
Enter the second number: 56
Enter an operator (+, -, *, /): +
The result is: 79.0
```

```
===== RESTART: D:/balli/cs/11th Class_2
Enter the first number: 45
Enter the second number: 89
Enter an operator (+, -, *, /): -
The result is: -44.0
```

```
===== RESTART: D:/balli/cs/11th Class_2
Enter the first number: 67
Enter the second number: 89
Enter an operator (+, -, *, /): *
The result is: 5963.0
```

```
===== RESTART: D:/balli/cs/11th Class_2
Enter the first number: 78
Enter the second number: 3
Enter an operator (+, -, *, /): /
The result is: 26.0
```



**Program 4:** Write a program to create a simple calculator to perform basic arithmetic operations on two numbers. The program should do the following:

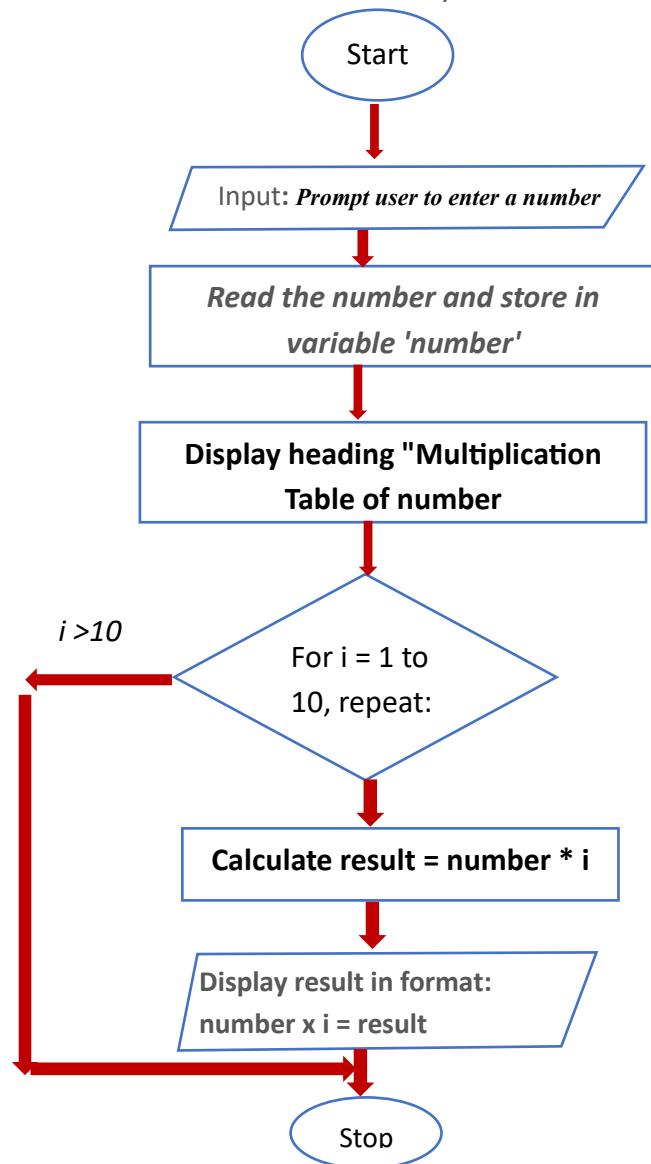
- d. Accept two numbers from the user.
- e. Ask user to input any of the operator(+,-,\*,/). An error message is displayed if the user enters anything else.
- f. As per the input of operator, do the corresponding operation on numbers entered by user.

```
def main():  
    num1 = float(input("Enter the first number: "))  
    num2 = float(input("Enter the second number: "))  
  
    # Ask for an operator  
    operator = input("Enter an operator (+, -, *, /): ")  
  
    # Perform the corresponding operation  
    if operator == '+':  
        result = num1 + num2  
        print(f"The result is: {result}")  
    elif operator == '-':  
        result = num1 - num2  
        print(f"The result is: {result}")  
    elif operator == '*':  
        result = num1 * num2  
        print(f"The result is: {result}")  
    elif operator == '/':  
        result = num1 / num2  
        print(f"The result is: {result}")  
  
if __name__ == "__main__":  
    main()
```

**Program 1:** Write a program to print the table of a given number. The number has to be entered by the user.

### Algorithm: Print Multiplication Table of a Given Number

1. Start
2. Prompt the user to enter a number.
3. Read the number and store it in a variable number.
4. Display a heading: "Multiplication Table of number".
5. Repeat the following steps from  $i = 1$  to  $i = 10$ :
  - o Calculate result = number \* i
  - o Display the result in the format: number  $\times$  i = result
6. End



Enter a number to print its multiplication table: 8

Multiplication Table of 8:

```
8 x 1 = 8
8 x 2 = 16
8 x 3 = 24
8 x 4 = 32
8 x 5 = 40
8 x 6 = 48
8 x 7 = 56
8 x 8 = 64
8 x 9 = 72
8 x 10 = 80
```

**Program 1:** Write a program to print the table of a given number. The number has to be entered by the user.

```
def main():
    # Ask the user to enter a number
    number = int(input("Enter a number to print its multiplication table: "))

    print(f"\nMultiplication Table of {number}:\n")

    # Print the multiplication table from 1 to 10
    for i in range(1, 11):
        print(f"{number} x {i} = {number * i}")

# Call the main function
if __name__ == "__main__":
    main()
```

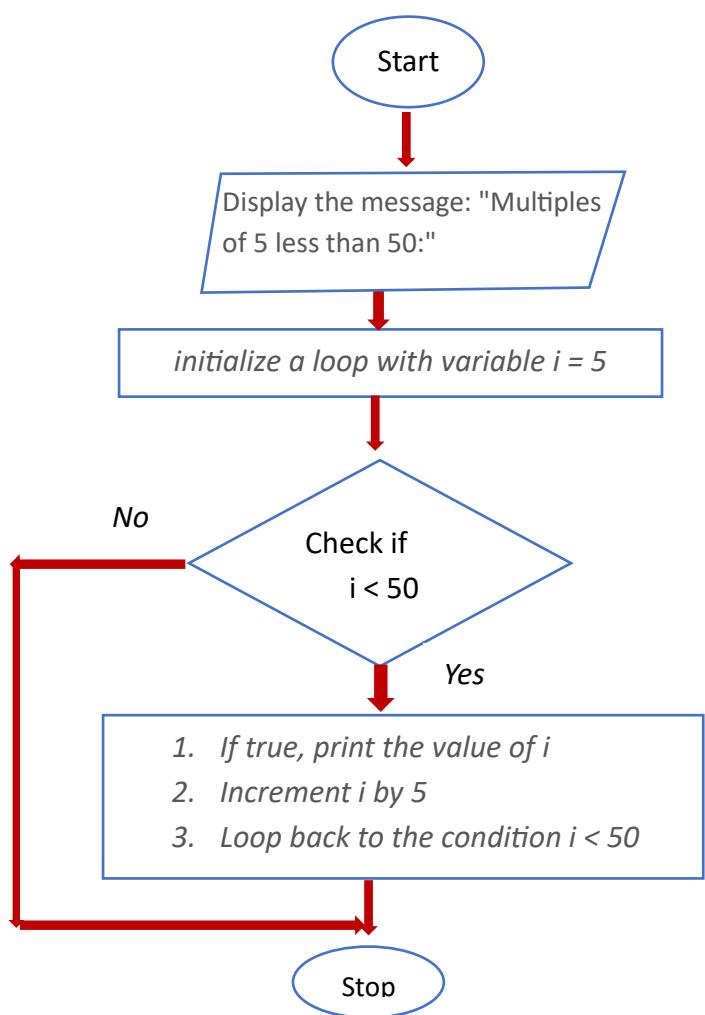
**Program 2:** Write a program to print all multiple of 5 that are smaller than 50 with the help of range function with for loop

### Algorithm: Print Multiples of 5 Less Than 50

1. **Start**
2. **Display** the message: "Multiples of 5 less than 50:"
3. Set the loop to start at 5 and go up to (but not including) 50, incrementing by 5
4. **Repeat** the following steps for each value  $i$  in the range 5 to 45:
  - o **Print** the value of  $i$
5. **End**

Multiples of 5 less than 50:

```
5
10
15
20
25
30
35
40
45
```



**Program 2:** Write a program to print all multiple of 5 that are smaller than 50 with the help of range function with for loop

```
def main():
    print("Multiples of 5 less than 50:\n")
    # Use range with a step of 5, starting from 5 up to (but not including) 50
    for i in range(5, 50, 5):
        print(i)
    # Call the main function
if __name__ == "__main__":
    main()
```

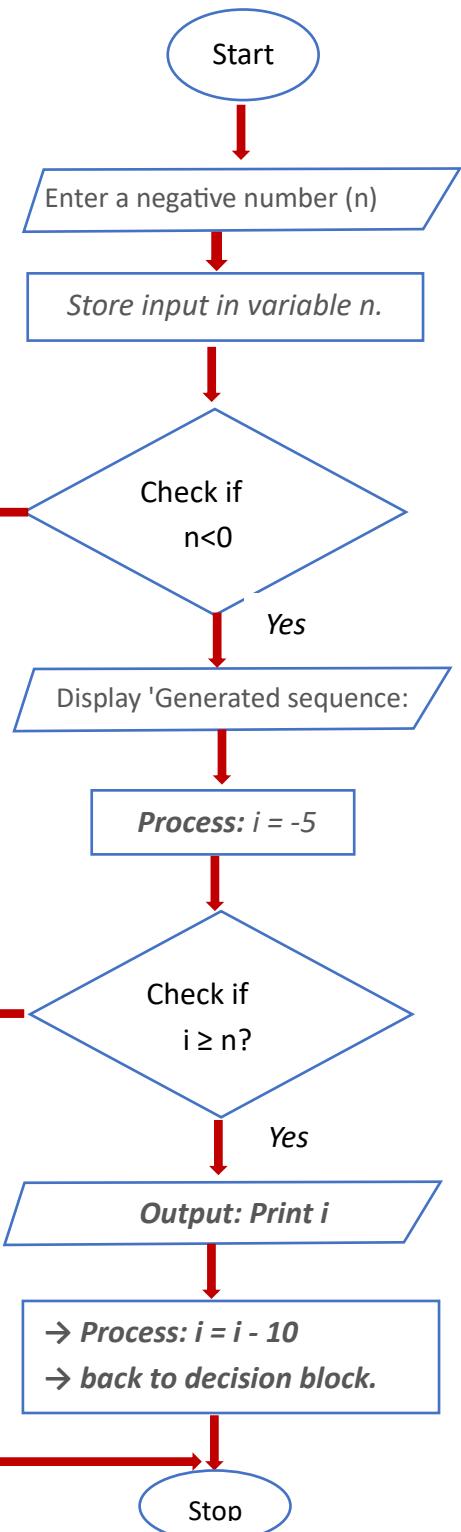
**Program 3:** Write a program to generate the sequence :

-5,-15,-25..... upto n

Where n is a negative number input by the user.

### Algorithm: Generate Sequence -5, -15, -25, ..., up to n

1. **Start**
2. **Prompt** the user to enter a negative number n.
3. **Read** the input and store it in variable n.
4. **If** n is not negative, **display** an error message and **end** the program.
5. **Display** the message: "Generated sequence:"
6. **Initialize** a loop starting from -5, decreasing by 10, until the value is less than or equal to n:
  - o **Print** the current value.
7. **End**



Enter a negative number (end of sequence) : -98

Generated sequence:  
-5, -15, -25, -35, -45, -55, -65, -75, -85, -95,

Enter a negative number (end of sequence) : 56  
Please enter a negative number.

Program 3: Write a program to generate the sequence :

-5,-15,-25..... upto n

Where n is a negative number input by the user.

```
def main():
    # Ask the user to enter a negative number
    n = int(input("Enter a negative number (end of sequence):"))

    # Check if input is negative
    if n >= 0:
        print("Please enter a negative number.")
        return

    print("\nGenerated sequence:")

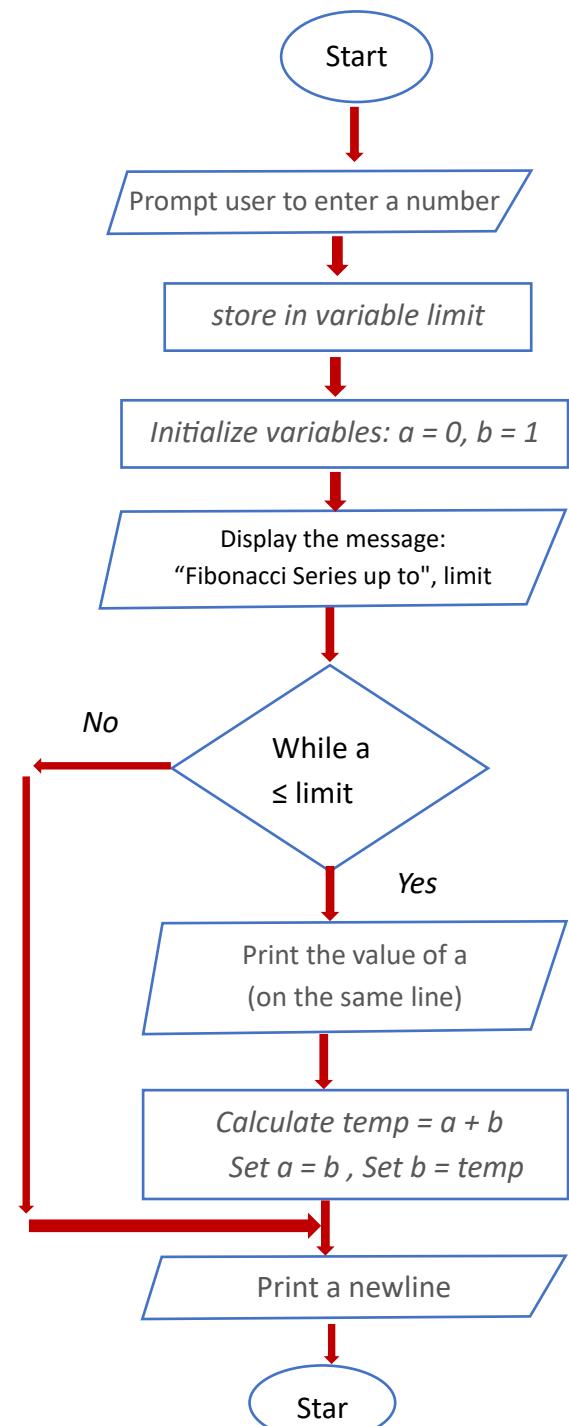
    # Start from -5 and decrement by 10 until reaching or passing n
    for i in range(-5, n - 1, -10):
        print(i, end=", ")

# Call the main function
if __name__ == "__main__":
    main()
```

**Program 4:** Write a program to print the Fibonacci Series up to a number entered by the user.

### Algorithm: Print Fibonacci Series up to a Given Number

1. **Start**
2. Prompt the user to **enter a number** and **store it in limit**
3. Initialize two variables:  
     $\rightarrow a \leftarrow 0$   
     $\rightarrow b \leftarrow 1$
4. Display the message: "Fibonacci Series up to", limit, ":"
5. **Repeat while a is less than or equal to limit:**
  1. Print the value of a (on the same line)
  2. Compute the next Fibonacci number:  
 $\rightarrow temp \leftarrow a + b$
  3. Update values:  
 $\rightarrow a \leftarrow b$   
 $\rightarrow b \leftarrow temp$
6. Print a newline (for formatting)
7. **End**



```
===== RESTART: D:/balli/cs/11th Class_2025-26/Chapter 4_Python/Program_G_4.py
Enter a number to print Fibonacci series up to: 50
Fibonacci Series up to 50 :
0 1 1 2 3 5 8 13 21 34
```

```
===== RESTART: D:/balli/cs/11th Class_2025-26/Chapter 4_Python/Program_G_4.py
Enter a number to print Fibonacci series up to: 236
Fibonacci Series up to 236 :
0 1 1 2 3 5 8 13 21 34 55 89 144 236
```

Program 4: Write a program to print the Fibonacci Series up to a number entered by the user.

```
def main():
    # Program to print the Fibonacci series up to a given number

    # Get the upper limit from the user
    limit = int(input("Enter a number to print Fibonacci series up to: "))

    # Initialize the first two Fibonacci numbers
    a, b = 0, 1

    print("Fibonacci Series up to", limit, ":")

    # Print Fibonacci numbers while the next number is less than or equal to the limit
    while a <= limit:
        print(a, end=' ')
        a, b = b, a + b # Update values

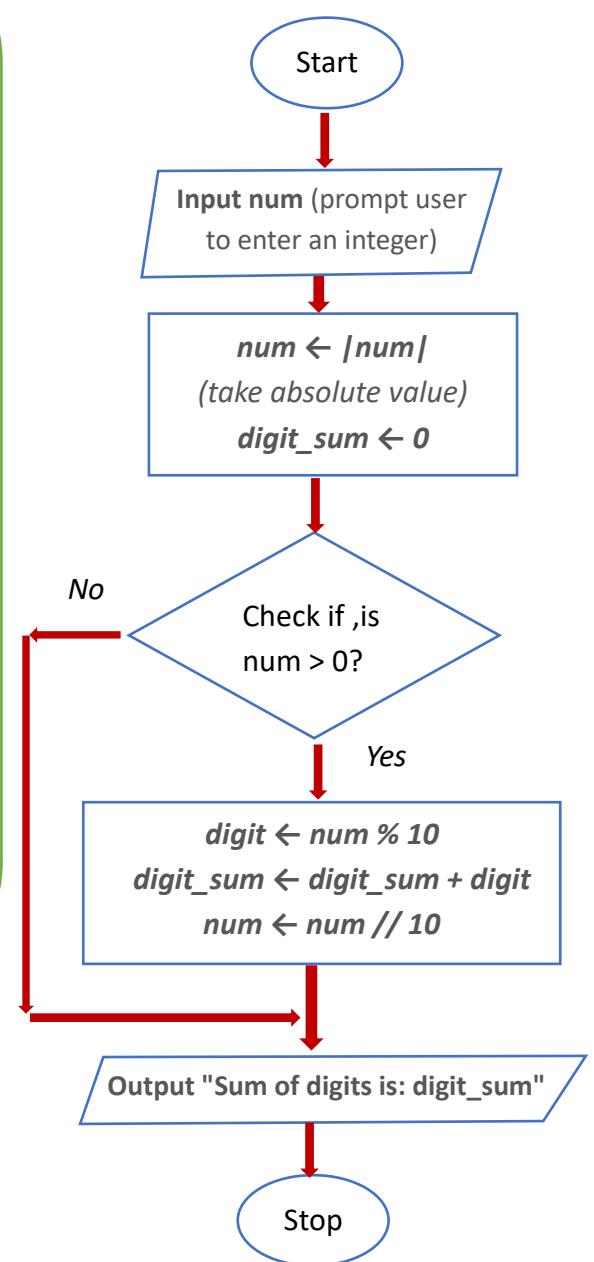
    print() # Newline after the series

# Call the main function
if __name__ == "__main__":
    main()
```

**Program 5:** Write a program to print the sum of digit of an integer number, input by the user.

### Algorithm: Sum of Digits of an Integer

1. Start
2. Prompt the user to **enter an integer number** and store it in num
3. Take the **absolute value** of num to handle negative input (i.e.,  $\text{num} \leftarrow |\text{num}|$ )
4. Initialize  $\text{digit\_sum} \leftarrow 0$
5. **Repeat while**  $\text{num} > 0$ :
  1.  $\text{digit} \leftarrow \text{num} \% 10$   
# Extract the last digit
  2.  $\text{digit\_sum} \leftarrow \text{digit\_sum} + \text{digit}$   
# Add digit to the sum
  3.  $\text{num} \leftarrow \text{num} // 10$   
# Remove the last digit
6. Print "Sum of the digits is: ",  $\text{digit\_sum}$
7. End



```
===== RESTART: D:/balli/cs/11th Class_2025-26/Chapter 4_Python/Program_G_5.py =
Enter an integer number: 245
Sum of the digits is: 11
```

```
===== RESTART: D:/balli/cs/11th Class_2025-26/Chapter 4_Python/Program_G_5.py =
Enter an integer number: 5678
Sum of the digits is: 26
```

Program 5: Write a program to print the sum of digit of an integer number , input by the user.

```
def main():
    # Program to calculate the sum of digits of a number

    # Get the number from the user
    num = int(input("Enter an integer number: "))

    # Make sure we handle negative numbers too
    num = abs(num)

    # Initialize sum variable
    digit_sum = 0

    # Loop to extract and sum the digits
    while num > 0:
        digit = num % 10      # Get the last digit
        digit_sum += digit    # Add digit to the sum
        num = num // 10       # Remove the last digit

    # Print the result
    print("Sum of the digits is:", digit_sum)

# Call the main function
if __name__ == "__main__":
    main()
```

**Program 6:** Write a program to print Patterns.

### Algorithm for the Pattern Printing Program

#### 1. Start

#### 2. Set rows $\leftarrow 4$

#### Pattern 1: Increasing Asterisks

3. Print a heading: "Pattern 1"
4. Repeat for  $i$  from 1 to rows (inclusive):
  - o Print '\*' repeated  $i$  times

#### Pattern 2: Decreasing Asterisks

5. Print a heading: "Pattern 2"
6. Repeat for  $i$  from rows down to 1:
  - o Print '\*' repeated  $i$  times

#### Pattern 3: Increasing Number Triangle

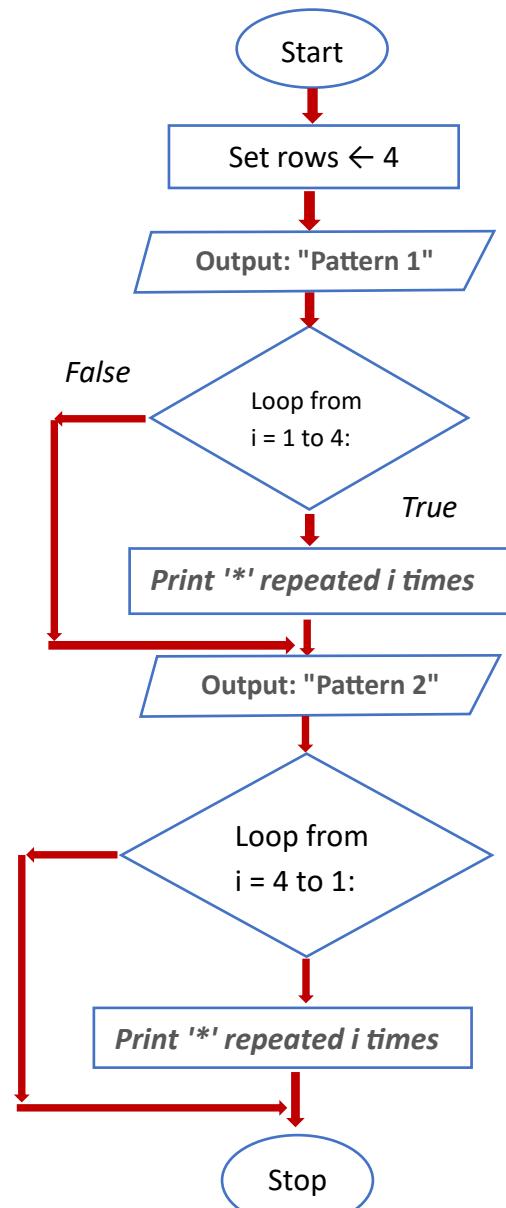
7. Print a heading: "Pattern 3"
8. Repeat for  $i$  from 1 to rows (inclusive):
  - o For each  $j$  from 1 to  $i$ :
    - Print  $j$  on the same line with a space
  - o Print a newline after each row

#### Pattern 4: Decreasing Number Triangle

9. Print a heading: "Pattern 4"
10. Repeat for  $i$  from rows down to 1:
  - o For each  $j$  from 1 to  $i$ :
    - Print  $j$  on the same line with a space
  - o Print a newline after each row

#### 11. End

Pattern 1	Pattern 2	Pattern 3	Pattern 4
*	****	1	1 2 3 4
**	***	1 2	1 2 3
***	**	1 2 3	1 2
****	*	1 2 3 4	1



Program 6: Write a program to print Patterns.

```
def main():
    # Input number of rows from user
    rows = 4
    # Loop to print pattern
    print( "\n\n\n\n Patten 1 \n ")
    for i in range(1, rows + 1):
        print('*' * i) #This line prints a string of asterisks (*), repeated i times.
        #In Python, 'string' * number repeats the string.
        print("\n\n\n\n Patten 2 \n ")
        for i in range(rows,0,-1):
            print('*' * i) #This line prints a string of asterisks (*), repeated i times.

        print("\n\n\n\n Patten 3 \n ")
        # Outer loop for rows
        for i in range(1, rows + 1):
            # Inner loop for numbers in each row
            for j in range(1, i + 1):
                print(j, end=' ')
            print() # Newline after each row
            print( "\n\n\n\n Patten 4 \n ")
        # Outer loop for rows
        for i in range(rows,0,-1):
            # Inner loop for numbers in each row
            for j in range(1, i + 1):
                print(j, end=' ')
            print() # Newline after each row

    # Call the main function
if __name__ == "__main__":
    main()
```

## Practical 1: Creating a Database

```
mysql> create database labactivity;
Query OK, 1 row affected (0.07 sec)

mysql> show databases
-> ;
+-----+
| Database |
+-----+
| information_schema |
| labactivity |
| mysql |
| performance_schema |
| sakila |
| sys |
| world |
+-----+
7 rows in set (0.04 sec)

mysql> use labactivity;
Database changed
mysql> create table practical(sid int, subject varchar(25), period int, topic varchar(30), minute int);
Query OK, 0 rows affected (0.14 sec)
```

## Practical 2: Insert Records into Table

```
mysql> INSERT INTO PRACTICAL VALUES (1001, "COMPUTER SCIENCE", 7, "MSWORD", 35);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM PRACTICAL;
+-----+-----+-----+-----+
| sid | subject | period | topic | minute |
+-----+-----+-----+-----+
| 1001 | COMPUTER SCIENCE | 7 | MSWORD | 35 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

## Program 3: Insert Partial Data

```
mysql> INSERT INTO PRACTICAL(SID, SUBJECT) VALUES (1004, "DIGITAL ELECTRONICS");
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM PRACTICAL;
+-----+-----+-----+-----+
| sid | subject | period | topic | minute |
+-----+-----+-----+-----+
| 1001 | COMPUTER SCIENCE | 7 | MSWORD | 35 |
| 1004 | DIGITAL ELECTRONICS | NULL | NULL | NULL |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

## Chapter 05: Basic Concepts of Database Management

### Practical 1: Create Database and Table

1. CREATE DATABASE LABACTIVITY;
2. SHOW DATABASES;
3. USE LABACTIVITY;
4. CREATE TABLE PRACTICAL ( SID INT, SUBJECT VARCHAR(25),  
PERIOD INT, TOPIC VARCHAR(30), MINUTE INT );

### Practical 2: Insert Records into Table

5. INSERT INTO PRACTICAL VALUES (1001, "COMPUTER SCIENCE", 7, "MSWORD", 35);
6. SELECT \* FROM PRACTICAL;
7. INSERT INTO PRACTICAL VALUES (1002, "IT/ITES", 4, "PAINT", 20),  
(1003, "COMPUTER APPLICATION", 2, "PYTHON", 38);
8. SELECT \* FROM PRACTICAL;

### Program 3: Insert Partial Data

9. INSERT INTO PRACTICAL(SID, SUBJECT) VALUES (1004, "DIGITAL ELECTRONICS");
10. SELECT \* FROM PRACTICAL;

## Practical 4: Alter Table and Add Column

```
mysql> ALTER TABLE PRACTICAL ADD COLUMN CLASS VARCHAR(10);
```

Query OK, 0 rows affected (0.11 sec)

Records: 0 Duplicates: 0 Warnings: 0

```
mysql> SELECT * FROM PRACTICAL;
```

sid	subject	period	topic	minute	CLASS
1001	COMPUTER SCIENCE	7	MSWORD	35	NULL
1004	DIGITAL ELECTRONICS	NULL	NULL	NULL	NULL
1004	DIGITAL ELECTRONICS	NULL	NULL	NULL	NULL

3 rows in set (0.00 sec)

## Practical 5: Update Data with Condition

```
mysql> UPDATE PRACTICAL
```

```
    -> SET CLASS = "11TH"
```

```
    -> WHERE CLASS IS NULL;
```

Query OK, 3 rows affected (0.01 sec)

Rows matched: 3 Changed: 3 Warnings: 0

```
mysql>
```

```
mysql> SELECT * FROM PRACTICAL;
```

sid	subject	period	topic	minute	CLASS
1001	COMPUTER SCIENCE	7	MSWORD	35	11TH
1004	DIGITAL ELECTRONICS	NULL	NULL	NULL	11TH
1004	DIGITAL ELECTRONICS	NULL	NULL	NULL	11TH

3 rows in set (0.00 sec)

## Practical 6: Update Specific Row

```
mysql> UPDATE PRACTICAL SET PERIOD = 1, TOPIC = "EXCEL", MINUTE = 30 WHERE SID = 1004;
```

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

```
mysql> SELECT * FROM PRACTICAL;
```

sid	subject	period	topic	minute	CLASS
1001	COMPUTER SCIENCE	7	MSWORD	35	11TH
1004	DIGITAL ELECTRONICS	1	EXCEL	30	NULL

2 rows in set (0.00 sec)

## Chapter 05: Basic Concepts of Database Management

### Practical 4: Alter Table and Add Column

11. ALTER TABLE PRACTICAL ADD COLUMN CLASS VARCHAR(10);  
12. SELECT \* FROM PRACTICAL;

### Practical 5: Update Data with Condition

13. UPDATE PRACTICAL SET CLASS = "11TH" WHERE CLASS IS NULL;  
14. SELECT \* FROM PRACTICAL;

### Practical 6: Update Specific Row

15. UPDATE PRACTICAL SET PERIOD = 1, TOPIC = "EXCEL", MINUTE = 30 WHERE SID = 1004;  
16. SELECT \* FROM PRACTICAL;

## Practical 7: Display Selected Columns

```
mysql> SELECT SID, SUBJECT, MINUTE FROM PRACTICAL;
+----+-----+-----+
| SID | SUBJECT           | MINUTE |
+----+-----+-----+
| 1001 | COMPUTER SCIENCE |      35 |
| 1004 | DIGITAL ELECTRONICS |      30 |
+----+-----+-----+
2 rows in set (0.00 sec)
```

## Practical 8: Conditional Data Retrieval

```
mysql> SELECT SID, SUBJECT, TOPIC FROM PRACTICAL WHERE MINUTE <= 30;
+----+-----+-----+
| SID | SUBJECT           | TOPIC |
+----+-----+-----+
| 1004 | DIGITAL ELECTRONICS | EXCEL |
+----+-----+-----+
1 row in set (0.00 sec)
```

## Practical 9: Sort Records by Minute Ascending

```
mysql> SELECT * FROM PRACTICAL ORDER BY MINUTE;
+----+-----+-----+-----+-----+-----+
| sid | subject           | period | topic | minute | CLASS |
+----+-----+-----+-----+-----+-----+
| 1004 | DIGITAL ELECTRONICS |      1 | EXCEL |      30 | NULL   |
| 1001 | COMPUTER SCIENCE |      7 | MSWORD |      35 | 11TH   |
+----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

## Practical 10: Sort Records by Minute Descending

```
mysql> SELECT * FROM PRACTICAL ORDER BY MINUTE DESC;
+----+-----+-----+-----+-----+-----+
| sid | subject           | period | topic | minute | CLASS |
+----+-----+-----+-----+-----+-----+
| 1001 | COMPUTER SCIENCE |      7 | MSWORD |      35 | 11TH   |
| 1004 | DIGITAL ELECTRONICS |      1 | EXCEL |      30 | NULL   |
+----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

## Chapter 05: Basic Concepts of Database Management

### Practical 7: Display Selected Columns

17. SELECT SID, SUBJECT, MINUTE FROM PRACTICAL;

### Practical 8: Conditional Data Retrieval

18. SELECT SID, SUBJECT, TOPIC FROM PRACTICAL WHERE MINUTE <= 30;

### Practical 9: Sort Records by Minute Ascending

19. SELECT \* FROM PRACTICAL ORDER BY MINUTE;

### Practical 10: Sort Records by Minute Descending

20. SELECT \* FROM PRACTICAL ORDER BY MINUTE DESC;

# STEM from Home

## Information Technology

### Learning to Code

In this era of information technology and computing, anyone with even a slight interest in the world of computers must have come across the word 'coding'. Coding involves writing instructions (codes) for creating a software program that solves any problem, be it an application, website, or game.

Most of you must be aware that computers understand strings of ones and zeroes known as the binary language. In order to make a computer work as per our instructions, we need to speak in its own language. Coding is basically the task of translation from human language to a language that computers or electronic machines understand. The person responsible of carrying out this task is called a coder. As computer technology has become very important, the number of opportunities in this field as a coder continues to grow.

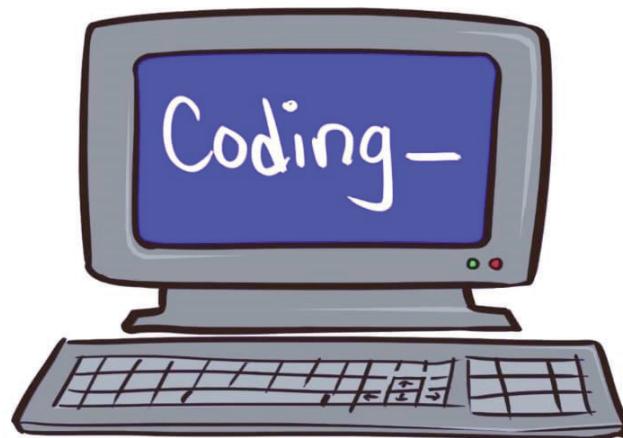
In order to become a coder, you need to be able to write code in at least one or more programming languages such as Python, C or Java. With knowledge of coding in any of the programming languages, you are able to write code, providing necessary information to the computer to make it work according to your instructions.

A simple text editor and a compiler or interpreter would suffice for the purpose of coding. Visit the following link to know more about interpreter and compiler.

<https://www.geeksforgeeks.org/language-processors-assembler-compiler-and-interpreter/>

Today we are going to learn some coding concepts by writing few lines of code in a popular programming language called Python. While writing the code, we shall also learn to solve a real-world problem of encryption and decryption.

Encryption is converting normal message into a secret message that no one is able to understand except the person to whom it is sent. While decryption is the process of converting this secret message into its original form.



*"Learning to code is useful no matter what your career ambitions are."*

—Arianna Huffington, Founder, The Huffington Post

# Main Activity: Learn to Code – Encrypting a Message

## Introduction

In this activity you will develop a program in Python that will convert a simple meaningful message into something that cannot be understood.



## What You Will Need

1. A computer connected with high-speed internet
2. A modern browser like Microsoft Edge or Google Chrome
3. An account at trinket.io

To understand how to start working on trinket.io, go through the [Getting Started with Trinket](#) document.

## Bonus Activities

### Activity 1: Learn to Code – Decrypting a Message

#### Introduction

You have learnt the basic concepts of coding and have written a code to encrypt a message. Suppose you send some encrypted messages to your friend. He receives your message, and is able to read it, but unable to understand the meaning of it. Why? Because he does not have a program that can decrypt it. Your task in this activity is to develop a program for your friend so that he can decrypt your message.

#### What You Will Need

- A computer connected with high-speed internet
- A modern browser like Microsoft Edge or Google Chrome
- An account at trinket.io

## What You Will Learn

- How to use trinket.io for coding and testing the programs
- Basics concepts and constructs of coding in Python like:
  - ▶ Defining variables
  - ▶ Using assignment, incremental assignment, and modulo operators
  - ▶ Accepting user inputs
  - ▶ Using programming constructs like ‘for’ loops and ‘if-then-else’ statements
  - ▶ Displaying outputs

Visit the following links to get more clarity on incremental assignment and modulo operators.

[The Python Modulo Operator](#)

[Python += Operator](#)

## Getting Started

You can access [guidelines for the activity here](#).

Access the [completed activity by clicking here](#).



To understand how to start working on trinket.io, go through the [Getting Started with Trinket](#) document.

## What You Will Learn

How to reuse and modify the coding modules  
Getting Started  
Access detailed [guidelines for the activity here](#).

# Challenge Activity: Learn to Code – Enhance the Encryption Level

Your task in this activity is to make changes in the encryption code to enhance encryption level so that it is more difficult for an unauthorised person to decode your message.

Test your code for encryption by typing a message which has more than two words. Provide a key and encrypt it.

Note down the original message, key, and the encrypted message.

You will observe that all the letters of the original message have been replaced with different letters. But the spaces are as it is. There is no encryption for the spaces. E.g., if the original message is “*i am a coder*”, key is 2, then the result of encryption should be “*k co c eqfgt*”.

Powered by  trinket

Please enter a message: i am a coder

Enter a key (1-26) : 2

Your new message is: k co c eqfgt

Make the necessary modification in the code so that it replaces the spaces in the message with a specific character. For example, if the original message is “*i am a coder*”, key is 2, then the result of encryption should be “*k\*co\*c\*eqfgt*”.

Powered by  trinket

Please enter a message: i am a coder

Enter a key (1-26) : 2

Your new message is: k\*co\*c\*eqfgt

Work independently.

Best of luck!