➢ Exercise 18: while Loop

Problem:
Write a Python program that asks the user for a number and calculates the factorial of that number using a while loop.

Practical 4: Basic ATM Simulation

Problem Statement:

Create a Python program that acts as a basic ATM. The program should:

1. Display an initial balance (e.g., ₹5000).
2. Allow the user to withdraw money, ensuring they do not withdraw more than the available balance.
3. Use a loop to allow multiple transactions until the user exits.
4. Deduct the amount and display the remaining balance after each transaction.
5. If the balance goes below ₹100, warn the user about a low balance.

Practical:3 University Admission Eligibility Check

Problem Statement:

Write a Python program that checks if a student is eligible for university admission. The program should:

1. Ask the user to enter their percentage marks in three subjects.
2. Calculate the average marks.
3. If the average is above 75%, print "**Eligible for Admission**"; otherwise, print "**Not Eligible**".
4. If the student scores above 90%, display a message saying "**You qualify for a scholarship!**".
5. Handle edge cases (e.g., if the user enters marks below 0 or above 100, display an error message).

➢ Exercise 17: Nested if-else Statement

Problem: Write a Python program that asks the user for a year and checks whether it is a leap year or not. A year is a leap year if:

- It is divisible by 4, and
- If it is a century year (ending in 00), it must also be divisible by 400.

➢ Exercise 16: if-elif Statement

Problem: Write a Python program that asks the user for the temperature in Celsius and classifies it as "Cold", "Warm", or "Hot" based on the following conditions:

- Below 15°C: Cold
- 15°C to 30°C: Warm
- Above 30°C: Hot

➢ Exercise 15: if-else Statement

Problem: Write a Python program that asks the user for their exam score. If the score is 40 or more, print "Pass", otherwise print "Fail".

➢ Exercise 14: if Statement

Problem: Write a Python program that asks the user for their age. If the age is 18 or above, print "**You are eligible to vote**".

Practical: 2 Automated Billing System for a Coffee Shop

Problem Statement: Design a Python program that simulates an automated billing system for a coffee shop. The program should:

1. Display a menu with prices (e.g., Coffee - ₹70, Tea - ₹50, Sandwich - ₹100).
2. Ask the user to enter the item name and quantity.
3. Calculate the total bill using arithmetic operators. Apply 18% GST on the total. Display final bill amount after tax. (For computation, do not use if statements or any other concepts beyond what has been covered so far.)
4. Display the current date and time when generating the bill.
5. Format and display the bill in a structured manner using string manipulation.

➢ Exercise 13: Bitwise and Identity Operators

Problem Statement: Develop a Python program that:

1. Performs bitwise AND, OR, and XOR operations on two user-provided integers.
2. Verifies if two variables point to the same object in memory.

➢ Exercise 12: Arithmetic and Logical Operators

Problem Statement : Write a Python program that takes three numbers as input from the user. Perform the following operations:

1. Compute the sum, product, and difference of the numbers.
2. Check if all the numbers are positive using a logical operator.
3. Determine if the sum of the numbers is greater than 100.

Solution

➢ Exercise 11: Formatting and Calculations with Date and Time

Problem Statement: Write a Python program to perform the following:

1. Display the current date and time in the format: Weekday, Month Day, Year Hour:Minute:Second.
2. Calculate the number of days left until the user-specified date (e.g., New Year).
3. Add 45 days to the current date and display the result in YYYY-MM-DD format.

Practical 1: Collecting and Displaying Student Details

Problem Statement: Write a Python program that **Problem Statement:** Write a Python program that collects basic student details and displays them in a structured format. The program should:

1. Ask the user to enter their **full name, age, course preference, and expected graduation year**.
2. Store the data in **variables**.
3. Use **arithmetic operations** to calculate the number of years left until graduation.
4. Print the details in a **well-formatted output**, displaying:

```
------ Student Enrollment Form ------
Name: Patel Harsh
Age: 18
Course: Computer Engineering
Expected Graduation Year: 2028
Years Left Until Graduation: 4
---------------------------------------
```

➢ Exercise 9: Arithmetic Operations

Problem Statement: Write a Python program that performs the following operations on two user-provided numbers:

1. Calculates and prints their sum, difference, product, and division result.
2. Finds and displays the remainder when the first number is divided by the second.
3. Calculates and prints the result of raising the first number to the power of the second.

➢ Exercise 10: Type Conversion and Complex Numbers

Problem Statement: Write a Python program to:

1. Convert a floating-point number to an integer and vice versa.
2. Create a complex number from two user inputs (real and imaginary parts) and display its real and imaginary components.
3. Calculate the conjugate of the complex number.

➢ Exercise 8: Dictionary Operations

Problem Statement: Define a dictionary with at least three key-value pairs and display its keys and values.

➢ Exercise 7: Sum of List Elements

Problem Statement: Create a list of numbers and calculate their sum using the sum() function.

➢ Exercise 6: String to Integer and Float Conversion

Problem Statement: Convert a string containing a number (e.g., "45") into an integer and a float.

➢ Exercise 5: Check Variable Types

Problem Statement: Write a program to check the type of a variable using type().

➢ Exercise 4: Create Variables of Different Types

Problem Statement: Create variables of different data types (int, float, str, list, tuple, set, dict, bool) and print their values.

➢ Exercise 3: Use of Multi-line Comments

Problem Statement: Write a Python program to demonstrate the use of multi-line comments.

➢ Exercise 4: Create Variables of Different Types

Problem Statement: Create variables of different data types (int, float, str, list, tuple, set, dict, bool) and print their values.

➢ Exercise 1: Display a Message

Problem Statement: Write a Python program to display the message "Hello, Python!".

➢ Exercise 2: Input Name and Age

Problem Statement: Create a program that asks the user for their name and age, then prints a message using the input.