Assignment 02

Question 1: Variables and Data Types

Problem: Write a Python program that:

1. Accepts a string, an integer, a float, and a boolean from the user.
2. Initializes variables for each type, and prints them out.
3. Convert the string to uppercase and print it.
4. Check if the integer is even or odd and print the result.
5. Multiply the float by 2 and print the result.

Example Input:

Enter a string: python

Enter an integer: 25

Enter a float: 3.14

Enter a boolean (True/False): True

Example Output:

Uppercase String: PYTHON

The number 25 is Odd

Doubled float: 6.28

Question 2: Operators

Problem: Write a Python program that:

1. Accepts two numbers as input from the user.
2. Performs and prints the result of all the arithmetic operations (addition, subtraction, multiplication, division, modulus, flow division) between these two numbers.

3. Use comparison operators to check if the first number is greater than the second, and if they are equal.
4. Use logical operators to combine two conditions (e.g., the first number is greater than the second, and the second number is less than 10).

Example Input:

Enter the first number: 10

Enter the second number: 3

Example Output:

Addition: 13

Subtraction: 7

Multiplication: 30

Division: 3.3333333333333335

Modulus: 1

Flow Division: 3

First number is greater than second: True

First number is equal to second: False

Both conditions are true: True

Question 3: Loops

Problem: Write a Python program that:

1. Accepts a list of integers from the user.
2. Loops through the list and prints out each number.
3. If a number is greater than 10, skip it using the continue statement.
4. Stop the loop if the number is 20 using the break statement.
5. After the loop ends, print a message that the loop ended naturally.

Example Input:

Enter a list of numbers separated by spaces: 5 10 12 15 20 8

Example Output:

5
10
Skipping 12
15
Breaking at 20
Loop ended naturally

Grading Criteria (Total: 30 Marks)

Criteria	Marks	Description
Correctness of Program (Logic)	15	The program correctly solves the problem by producing the expected output.
Code Quality (Readability)	5	Code is well-structured, follows Pythonic practices, and includes meaningful variable names.
Use of Concepts (Variables, Operators, Loops)	5	Proper use of variables, operators, loops, and control flow statements.
Testing & Input Handling	5	The program handles user inputs correctly and includes basic validation for edge cases.