

Output :

Enter first value ~~100~~ 150

Enter second value: 50

Addition: 200.00

Subtraction: 100.00

Multiplication: 7500

Division: 3

Date: 23-07-2025

Task 1: Running Python script and various expressions in an Interactive Interpreter.

(a) Perform Basic Mathematical Computations

Aim: To write a Python program that accepts two numerical inputs and perform addition, subtraction, multiplication, and division operations.

Algorithm

1. Start the program
2. Accept two numerical inputs from the user.
3. Perform:
  - Addition
  - Subtraction
  - Multiplication
  - Division (if second number is not zero)
4. Display the results.
5. End the program.

Program

```
num1 = float(input("Enter first value:"))  
num2 = float(input("Enter second value:"))  
print("Addition:", num1 + num2)  
print("Subtraction:", num1 - num2)  
print("Multiplication:", num1 * num2)  
print("Division:", num1 / num2)
```

Result: The program successfully performed all arithmetic operations on the given inputs and displayed the results.

Output:

Enter first score: 55

Enter second score: 70

$a > b$ : False

$a < b$ : True

$a == b$ : False

$a != b$ : True

$a >= b$ : False

$a <= b$ : True

## (b) Evaluate Relational Expressions

Aim: To develop a Python program that compares two numeric values using relational operations and displays the result of each comparison.

### Algorithm

1. Start the program
2. Accept two numbers from the user.
3. Apply the following relational operators:
  - Greater than ( $>$ )
  - Less than ( $<$ )
  - Equal to ( $==$ )
  - Not equal to ( $!=$ )
  - Greater than or equal to ( $>=$ )
  - Less than or equal to ( $<=$ )
4. Display the results.
5. End the program.

### Program

```
a = float(input("Enter first score:"))  
b = float(input("Enter second score:"))  
print("a > b:", a > b)  
print("a < b:", a < b)  
print("a == b:", a == b)  
print("a != b:", a != b)  
print("a >= b:", a >= b)
```

Result: The program correctly evaluated all the relational expressions between the two given inputs.

Output:

Enter marks for Test 1:

Enter marks for Test 2:

Enter marks for Test 3:

Passed all tests:

Passed at least one test:

Failed all tests:

1) Check logical conditions across Multiple Inputs.

Atm: To create a Python program that uses logical Operators (and, or, not) to evaluate conditions across three test scores.

Algorithm

1. Start the program
2. Accept three test scores from the user.
3. Use logical operators to evaluate:
  - If the candidate passed all tests (and)
  - If the candidate passed at least one test (or)
  - If the candidate failed all tests (not)
4. Display the results.
5. End the program.

Program

```
test 1 = int (input ("Enter marks for Test 1:"))
```

```
test 2 = int (input ("Enter marks for Test 2:"))
```

```
test 3 = int (input ("Enter marks for Test 3:"))
```

```
print ("Passed all tests:", test 1 > 40 and test 2 > 40 and  
test 3 > 40)
```

```
print ("Passed at least one test:", test 1 > 40 or test 2 > 40  
or test 3 > 40)
```

```
print ("Failed all tests:", not (test 1 > 40 or test 2 > 40  
or test 3 > 40))
```

Result: The program effectively evaluated logical expressions and correctly identified pass/fail conditions based on test scores.

VEL TECH - CSE	
EX NO.	
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	5