Assessment-02

Objectives

- Understand the different types of operators in a programming language (arithmetic, relational,
  logical, assignment, etc.).
- Explore how operator precedence and associativity govern the evaluation of complex expressions.
- Practice writing, predicting, and debugging expressions containing multiple operators.

Task-01: Basic Operator Usage

Aim:

Write a program that accepts two integer values and demonstrates the use of basic arithmetic operators like (+, −, ∗∗, //,) by displaying the result of each operation.

Sample Code:

```
a=int(input("Enter a number: "))
b=int(input("Enter another number: "))
print("Addition:",a,"+",b,"=",a+b)
print("Subtraction: ",a,"-",b,"=",a-b)
print("Multiplication: ",a,"*",b,"=",a*b)
print("Exponentiation: ",a,"**",b,"=",a**b)
print("Floor division: ",a,"//",b,"=",a//b)
```

Sample Output:

```
=============== RESTART: C:/Users/ATHARV BISHT/Desktop/task-01.py ==============
Enter a number: 10
Enter another number: 2
Addition: 10 + 2 = 12
Subtraction:  10 - 2 = 8
Multiplication:  10 * 2 = 20
Exponentiation:  10 ** 2 = 100
Floor division:  10 // 2 = 5
```

Task-02: Complex Expressions and Precedence

Aim:

Ask the user to input three integer values (a, b, c) and evaluate and print the result of expressions.

Sample Code:

```
a=int(input("Enter a number: "))
b=int(input("Enter another number: "))
c=int(input("Enter one more number: "))
print("1st expression: ", a+b*c-a+b*c)
print("2nd expression: ", (a+b)*c-(a/b)*c)
print("3rd expression: ", a*b/c+a*b/c)
print("4th expression: ", a+b-c*a/b-a+b-c*a/b)
```

Sample Output:

```
= RESTART: C:/Users/ATHARV BISHT/Desktop/task-01.py
Enter a number: 10
Enter another number: 5
Enter one more number: 2
1st expression:  20
2nd expression:  26.0
3rd expression:  50.0
4th expression:  2.0
```

Task-03: Predict and Verify Output

Aim:

Provide a set of code snippets containing complex expressions with mixed operators.

Sample Code:

```
# Snippet 1
# Predict: 10 + 3 * 2 -> multiplication first -> 10 + 6 = 16
print("Snippet 1:", 10 + 3 * 2)

# Snippet 2
# Predict: (10 + 3) * 2 -> parentheses first -> 13 * 2 = 26
print("Snippet 2:", (10 + 3) * 2)

# Snippet 3
# Predict: 20 / 2 ** 2 -> exponent first -> 20 / 4 = 5.0
print("Snippet 3:", 20 / 2 ** 2)

# Snippet 4
# Predict: 15 // 4 % 3 -> floor division first -> 15 // 4 = 3, then 3 % 3 = 0
print("Snippet 4:", 15 // 4 % 3)
```

Sample Output:

```
=============== RESTART: C:/Users/ATHARV BISHT/Desktop/task-01.py ==============
Snippet 1: 16
Snippet 2: 26
Snippet 3: 5.0
Snippet 4: 0
```

Task 4: Parentheses to control Precedence

Aim:

Demonstrate how inserting parentheses can change the result of expressions.

Sample Code:

```
a=float(input("Enter a number: "))
b=float(input("Enter another number: "))
c=float(input("Enter one more number: "))
print("1st expression: ", (a+b)*c/(a+b)**c)
print("2nd expression: ", a+(b-c)/a*(b/c))
print("3rd expression: ", a*(b/c)+(a+b)/c)
print("4th expression: ", a-(b+c)*c-(a+c))
```

Sample Output:

```
= RESTART: C:/Users/ATHARV BISHT/Desktop/task-01.py
Enter a number: 5
Enter another number: 2
Enter one more number: 1
1st expression:  1.0
2nd expression:  5.4
3rd expression:  17.0
4th expression:  -4.0
```

Sample Output (Without Parentheses):

```
=============== RESTART: C:/Users/ATHARV BISHT/Desktop/task-01.py ==============
Enter a number: 5
Enter another number: 2
Enter one more number: 1
1st expression:  7.4
2nd expression:  6.6
3rd expression:  17.0
4th expression:  0.0
```