21 May

**Python Basic - 2**

Q.1. Create two int type variables, apply addition, subtraction, division and multiplications and store the results in variables. Then print the data in the following format by calling the variables:

a= 10

b=5

print(f"Addition of two value is :{a+b}")

print(f"Addition of two value is :{a-b}")

print(f"Addition of two value is :{a\*b}")

print(f"Addition of two value is :{a/b}")

Q.2. What is the difference between the following operators:

(i) ‘/’ & ‘//’

/ and // operators are used for division, but they have different behaviors and produce different results.

* The / operator performs floating-point division. It divides the left operand by the right operand and returns the result as a floating-point number, even if the operands are integers. The result can have decimal places. For example

result = 7 / 2

print(result) # Output: 3.5

* // (Double Slash or Floor Division Operator):

The // operator performs floor division. It divides the left operand by the right operand and returns the largest integer less than or equal to the result. In other words, it discards the decimal part of the division result, resulting in an integer. For example:

result = 7 // 2

print(result) # Output: 3

(ii) ‘\*\*’ & ‘^’

* \*\* (Exponentiation Operator):

The \*\* operator is used for exponentiation or raising a number to a power. It

calculates the left operand raised to the power of the right operand. For example:

result = 2 \*\* 3

print(result) # Output: 8

* ^ (Bitwise XOR Operator):

The ^ operator is used for bitwise XOR (exclusive OR) operation. It performs a bitwise XOR operation between the binary representations of two operands. For example:

result = 5 ^ 3

print(result) # Output: 6

Q.3. List the logical operators.

We will understand all this logical operator by taking the below example

x = 5

y = 10

z = 3

**# Logical AND operator**

result\_and = (x > 0) and (y < 20)

print(result\_and) # Output: True

**# Logical OR operator**

result\_or = (x < 0) or (z > 5)

print(result\_or) # Output: True

**# Logical NOT operator**

result\_not = not (x == 5)

print(result\_not) # Output: False

Q.4. Explain right shift operator and left shift operator with examples.

Right shift operator: The right shift operator (>>) shifts the bits of the left operand to the right by the number of positions specified by the right operand. The rightmost bits are discarded, and the leftmost bits are filled with the sign bit (in case of signed integers) or with zeroes (in case of unsigned integers). For example:

Program:

x = 10 # Binary: 1010

result = x >> 2

print(result) # Output: 2 (Binary: 10)

Left shift operator: The left shift operator (<<) shifts the bits of the left operand to the left by the number of positions specified by the right operand. The rightmost bits are filled with zeroes, and the leftmost bits are discarded. For example:

Program:

x = 5 # Binary: 101

result = x << 2

print(result) # Output: 20 (Binary: 10100)

Q.5. Create a list containing int type data of length 15. Then write a code to check if 10 is present in the list or not.

List = [10,15,12,5,9,8,5,64,5,8,37,1,5,2,52]

print(len(List)) #Output: 15

if 10 in List:

print("Yes 10 is present in this list") #Output : Yes 10 is present in the list