**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:**

**First variable is \_\_ & second variable is \_\_.**

**Addition: \_\_ + \_\_ = \_\_**

**Subtraction: \_\_ - \_\_ = \_\_**

**Multiplication: \_\_ \* \_\_ = \_\_**

**Division: \_\_ / \_\_ = \_\_**

**Ans 1:**

a = 10

b = 2

sum = a + b

sub = a - b

mul = a \* b

div = a / b

print(f"First variable is {a} & second variable is {b}")

print(f"Addition: {a} + {b} = {sum}")

print(f"Subtraction: {a} - {b} = {sub}")

print(f"Multiplication: {a} \* {b} = {mul}")

print(f"Division: {a} / {b} = {div}")

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

**(i) ‘/’ & ‘//’**

**(ii) ‘\*\*’ & ‘^’**

**Ans 2:**

1. ‘/’ Performs floating-point division, always returning a floating-point result.

Eg: result = 7/2 # output 3.5

‘//’ Performs floor division, returning the integer result closest to zero.

Eg: result = 7/2 # output 3

1. ‘\*\*’ Exponentiation operator, used for raising a number to a power.

For eg : 2\*\*4 = 2 \* 2 \* 2 \* 2 = 16

‘^’ Bitwise XOR operator, used for bitwise manipulation of integers.  
For eg : 5 ^ 3 = 101 XOR 011 = 110 which is 6.

**Q.3. List the logical operators.**

**Ans 3:** Logical operators in python are :

1. ‘and’ : Returns TRUE if both the statements are TRUE
2. ‘or’ : Returns TRUE if any one statement is TRUE
3. ‘not’ : Returns TRUE is statement is FALSE and vise versa.

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

**Ans 4:** The right shift operator (>>) and the left shift operator (<<) are bitwise operators.

1. Right Shift Operator(>>): It shifts the bits of a number to the right by a specified number of positions.

Each right shift by one position effectively divides the number by 2.

Eg : x= 6 # binary 0b1000

y = x >> 2 # shifts right by 2 position

print(y) # output is 2 0b10

1. Leftt Shift Operator(<<): It shifts the bits of a number to the left by a specified number of positions.

Each left shift by one position effectively multiplies the number by 2.

Eg : x = 3 # binary 0b11

y = x << 2 # shifts left by 2 positions

print(y) # result 12 0b1100

**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.**

**Ans 5 :**

list\_val = [1,4,6,2,5,8,9,10,33,21,3,7,15,43,11]

if 10 in list\_val:

print("Present")

else:

print("Not present")