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: \_\_ / \_\_ = \_\_

first\_variable = 15

second\_variable = 5

addition = first\_variable + second\_variable

subtraction = first\_variable - second\_variable

multiplication = first\_variable \* second\_variable

division = first\_variable / second\_variable

print("First variable is {} & second variable is {}".format(first\_variable, second\_variable))

print("Addition: {} + {} = {}".format(first\_variable, second\_variable, addition))

print("Subtraction: {} - {} = {}".format(first\_variable, second\_variable, subtraction))

print("Multiplication: {} \* {} = {}".format(first\_variable, second\_variable, multiplication))

print("Division: {} / {} = {}".format(first\_variable, second\_variable, division))

Subtraction: 15 - 5 = 10

Multiplication: 15 \* 5 = 75

Division: 15 / 5 = 3.0

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

(i) ‘/’ & ‘//’

(ii) ‘\*\*’ & ‘^’

The operators / and // are both used for division in Python, but they have different results. The / operator divides the two operands and returns a floating-point number. The // operator divides the two operands and returns an integer, rounding down to the nearest integer.

The operators \*\* and ^ are both used for exponentiation in Python, but they have different results. The \*\* operator raises the first operand to the power of the second operand. The ^ operator performs bitwise exclusive OR on the two operands.

Q.3. List the logical operators.

AND (&&) - Returns true if both operands are true.

OR (||) - Returns true if either operand is true.

NOT (!) - Returns the opposite of the operand.

XOR (^) - Returns true if exactly one of the operands is true.

Ternary Operator (?:) - Returns one of two values, depending on the Boolean value of the first operand.

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

# Right shift operator

a = 10

b = a >> 2

print(b)

# Output: 2

# Left shift operator

a = 10

b = a << 2

print(b)

# Output: 40

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.

# Create a list containing int type data of length 15

list\_of\_numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]

# Check if 10 is present in the list or not

if 10 in list\_of\_numbers:

print("10 is present in the list")

else:

print("10 is not present in the list")

Output:

10 is present in the list