Python Basic – 2

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

# Create two int type variables

first\_variable = 10

second\_variable = 5

# Perform the operations

addition\_result = first\_variable + second\_variable

subtraction\_result = first\_variable - second\_variable

multiplication\_result = first\_variable \* second\_variable

division\_result = first\_variable / second\_variable

print("First variable is", first\_variable, "& second variable is", second\_variable)

print("Addition:", first\_variable, "+", second\_variable, "=", addition\_result)

print("Subtraction:", first\_variable, "-", second\_variable, "=", subtraction\_result)

print("Multiplication:", first\_variable, "\*", second\_variable, "=", multiplication\_result)

print("Division:", first\_variable, "/", second\_variable, "=", division\_result)

1. **What is the difference between the following operators:**
2. **‘/’ & ‘//’**
3. **(ii) ‘\*\*’ & ‘^’**

(ⅰ), Differences between ‘/’ and ‘//’:

**‘/’:** The forward slash is the standard division operator in python. When it is used with integers, it performs regular division and returns a floating-point result.

Result = 7/2

print(Result) #output: 3.5

**‘//’:** The double forward slash is known as floor division operator in python. It performs and returns the quotient, which is rounded down to nearest integer.

Result = 7//2

print(Result) #output: 3

(ⅱ), Differences between ‘\*\*’ and ‘^’:

**‘\*\*’:** The double asterisk is the exponentiation operator in python. It is used to raise a number to a certain power.

Result = 2 \*\* 3

print(Result) #output: 8

**‘^’:** The caret is used as bitwise XOR operator. It performs a bitwise XOR operation of two integers.

Result = 5 ^ 3

print(Result) #output: 6

1. **List the logical operators.**

Logical operators are used to perform logical operations on boolean values (True or False). In Python, the following are the logical operators:

1. Logical AND
2. Logical OR
3. Logical NOT

* **and**: The logical AND operator returns True if both operands are True, otherwise, it returns False. It requires both operands to be True for the overall expression to be True.
* **or:** The logical OR operator returns True if at least one of the operands is True. It returns False only when both operands are False.
* **not:** The logical NOT operator is a unary operator that negates the boolean value. If the operand is True, it returns False, and if the operand is False, it returns True.

These logical operators are widely used in conditionals and expressions to make decisions based on the truth values of certain conditions.

1. **Explain right shift operator and left shift operator with examples.**

The right shift and left shift are bitwise shift operators in Python. They are used to manipulate the bits of an integer by shifting them to the right or left respectively.

1. **Right shift(>>):** The right shift operator shifts the bits of a number to the right by a specified number of positions. For each shift, the rightmost bits are discarded, and new bits are filled in from the left with the same value as the original leftmost bit i.e., 0 for positive and 1 for negative numbers.

number = 12

shift = 2

result = number >> shift

print(result) #output: 3

1. **Left shift(<<)**: The left shift operator shifts the bits of a number to the left by a specified number of positions. For each shift, new 0 bits are filled in from the right.

number = 5

shift = 3

result = number << shift

print(result) #output: 40

1. **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

my\_list = [2, 5, 8, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65]

# Check if 10 is present in the list

if 10 in my\_list:

print("10 is present in the list.")

else:

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

In the above code snippet, we first create a list named my\_list containing 15 integer elements. Then, we use the **in** keyword to check if the value 10 is present in the list. If it is present, the code prints "10 is present in the list." Otherwise, it prints "10 is not present in the list." Since 10 is one of the elements in the list, the output shows that it is present.