**Assignment 7**

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:

n1=20

n2=5

sum=n1+n2

diff=n1-n2

mul=n1\*n2

div=n1/n2

print("First variable is",n1, "& the second variable is",n2)

print("Addition:",n1,"+",n2,"=",sum)

print("Subtraction:",n1,"-",n2,"=",diff)

print("Multiplication:",n1,"\*",n2,"=",mul)

print("Division:",n1,"/",n2,"=",div)

Output of the above code is:

First variable is 20 & the second variable is 5

Addition: 20 + 5 = 25

Subtraction: 20 - 5 = 15

Multiplication: 20 \* 5 = 100

Division: 20 / 5 = 4.0

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

Ans:

(i) ‘/’ operator performs division operation and returns the quotient as a floating number. Whereas ‘//’ operator returns an integer rounding the result towards –infinity (-∞).

Example:

print(15/4)

Output: 3.75

print(-15//4)

Output: -4

(ii) ‘\*\*’ operator is used to find the exponentiation i.e. to find certain power of a given number. Whereas, ‘^’ is a bitwise XOR operator. It converts the integers to their binary equivalent and performs ‘XOR’ operation on the binary numbers.

2\*\*3

Output: 8

5^8

Output: 13 (5 in binary = 0101, 8 in binary = 1000, XOR(0101,1000) = 1101 = 13

1. List the logical operators.

Ans: Logical or Boolean operators in Python are: and, or and not.

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

Ans: Right shift operator (>>) shifts the binary equivalent of an integer to the right by a specified number of positions.

Example:

x = 10

print(x>>2)

Output: 2.

Explanation: Binary equivalent of 10 is 1010. After getting shifted by two bits to the right, it becomes 0010 (=2 in decimal).

Left shift operator (<<) shifts the binary equivalent of an integer to the left by a specified number of positions.

x = 10

print(x<<2)

x = 10

print(x>>2)

Output: 40.

Explanation: Binary equivalent of 10 is 1010. After getting shifted by two bits to the left, it becomes 101000 (= 40 in decimal).

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.

Ans:

list1=[5, 8, 10, 15, 17, 11, -10, -5, -12, 32, 37, 67, 69, 85, 87]

if 10 in list1:

  print("The integer 10 is present in the list")

else:

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

Output:

The integer 10 is present in the list

list1=[5, 8, 13, 15, 17, 11, -10, -5, -12, 32, 37, 67, 69, 85, 87]

if 10 in list1:

  print("The integer 10 is present in the list")

else:

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

Output:

The integer 10 is not present in the list