

Day 4 25/11/25

02) Logical operation

Logical operations are used to combine multiple conditions and return either true or false.

There are 3 logical operations:-

- 1) And : Both condition need to be true then it returns true.
- 2) OR : Any one condition needs to be true.
- 3) NOT : It inverts the condition or returning false if true.

Ex:- a=10

b=15

Print (a==b) (and b!=a) : both are false

Print (a==b or b!=a)

Print (a!=b) (not a true value) true
(not b true value) true

Output : False

True

True

03) Comparison operation / Relational operators

Comparison operations are used to compare two values. They always return Boolean (true/false).

Comparison operations are

1) ==

2) !=

3) >

4) <

5) >=

6) <=

Ex:- $a = 10$

$b = 15$

Print ($a > b$)

Print ($a < b$)

Print ($a == b$)

Print ($a != b$)

Print ($a >= b$)

Print ($a <= b$)

Output :- False

True

False

True

False

True

Q4) Assignment operations

Assignment operations are used to assign value to variables.

Assignment operations are:-

$+ =$ (Add & Assign)

$- =$ (Sub & Assign)

$*$: (Multiply & Assign)

$/ =$ (Divide & Assign)

$\text{ff} =$ (Floor Divide & Assign)

$\% =$ (Modulus & Assign)

$*^ =$ (Power & Assign)

Ex: $a=10$

$b=15$

$a+b \# a=a+b$

Print (a)

$a*=b \# a=a*b$

Print (a)

$a-=b \# a=a-b$

Print (a)

$a|=b \# a=a|b$

Print (a)

$a**=b \# a=a**b$

Print (a)

$a^=b \# a=a^b$

Print (a)

Output : 25

10

150

10.0

0.0

0.0

0.0

05) Bitwise Operations

Bitwise operations are used to perform operations on the "Binary" (bit level) representations of numbers.

- * In computer very no is stored in P's & I's
- * Bitwise operation allow us to work directly on this basis

Binary Language

1	6	8	2	1	
$2^3(8)$	$2^2(4)$	$2^1(2)$	$2^0(1)$		
0	0	0	0	0	
0	0	0	1	1	
0	0	1	0	2	
0	0	1	1	3	
0	1	0	0	4	
0	1	0	1	5	
0	1	1	0	6	
0	1	1	1	7	
1	0	0	0	8	

Ex:- $11 \oplus 6 = 12$

$$\begin{array}{r} 11 \\ \downarrow \\ 1011 \end{array} \quad \begin{array}{r} 6 \\ \downarrow \\ 1100 \end{array}$$

$$\begin{array}{r} 1011 \\ 1100 \\ \hline 1000 \Rightarrow 8 \end{array}$$

$$\therefore 11 \oplus 6 = 8$$

Types of Bitwise operations are:-

i) And ($\&$)

$$a = 5$$

$$b = 3$$

Print ($a \& b$)

2) OR (|)

Print ($s / 3$)

$$\begin{array}{r} 0101 \\ 0011 \\ \hline 0111 = 7 \end{array}$$

3) XOR (^)

Print ($s ^ 3$)

$$\begin{array}{r} 0101 \\ 0011 \\ \hline 0110 = 6 \end{array}$$

4) NOT (~)

Print ($\sim s$)

Output = 6

5) Left Shift (<<)

Print ($s << 1$)

0101 → 1010

s becomes 10

6) Right Shift (>>)

Print ($s >> 1$)

0101 = 0010

s becomes 12

67 Identity operations

Identity operators are used to check whether two variables refer to the same object in memory.

1) is

2) is not

Ex: `a = [1, 2, 3]`

`b = a`

`Print (a is b)` (True)

Output `b = [1, 2, 3]`

`a = 8`

`b = 8`

`Print (a is b)` (False)

`Print (a is not b)`

Output : True

False

68>

Membership operation

Membership operations are used to check whether a value is in a sequence.

A sequence can be list, tuple, string, set etc..

1) in

2) not in

Ex :- $L1 = [4, 1, 3, 5]$

Print (5 in L1)

Print (3 not in L1)

Output : True

False

Example problem 03

Write a program to calculate, area of square, area of circle, area of triangle. take input from user.

```
=> s = float(input("Enter Square's side:"))
r = float(input("Enter the radius of circle:"))
b = float(input("Enter the base of triangle:"))
h = float(input("Enter the height of triangle:"))
```

$$\text{Area - Square} = s * s$$

$$\text{Radius - Circle} = 3.14159 * r * r$$

$$\text{Area - Triangle} = 0.5 * b * h$$

Print ("Area of Square")

Print ("Area of Circle")

Print ("Area of Triangle")