BCA – 502: Python Programming

Rahul Kumar Singh

In today's Class we have discussed on Python Identity operators and Python Bitwise operators.

Python Identity Operators:-

Identity operators compare the memory locations of two objects. There are two Identity operators explained below−

is operator:- Returns True if both variables are the same object.

Example:-

```
x = ["apple", "banana"]

y = ["apple", "banana"]

z = x

print(x is z)

# returns True because z is the same object as x

print(x is y)

# returns False because x is not the same object as y, even if they have the same content

print(x == y)
```

to demonstrate the difference betweeen "is" and "==": this comparison returns True because x is equal to y

Output:-

True

False

True

is not operator:- Returns True if both variables are not the same object.

Example:-

x = ["apple", "banana"]

y = ["apple", "banana"]

z = x

print(x is not z)

returns False because z is the same object as x

print(x is not y)

returns True because x is not the same object as y, even if they have the same content

print(x != y)

to demonstrate the difference betweeen "is not" and "!=": this comparison returns False because x is equal to y

Output:-

False

True

False

Q.)

```
#!/usr/bin/python
a = 10
b = 20
list = [1, 2, 3, 4, 5 ];

if ( a in list ):
   print "Line 1 - a is available in the given list"
else:
   print "Line 1 - a is not available in the given list"

if ( b not in list ):
   print "Line 2 - b is not available in the given list"
else:
   print "Line 2 - b is available in the given list"

a = 2
```

```
if ( a in list ):
   print "Line 3 - a is available in the given list"
else:
   print "Line 3 - a is not available in the given list"
```

Output:-

```
Line 1 - a is not available in the given list
Line 2 - b is not available in the given list
Line 3 - a is available in the given list
```

Python Bitwise Operators:-

Bitwise operators are used to compare (binary) numbers.

Some Bitwise Operators are as below:

& (AND) Operator:- Sets each bit to 1 if both bits are 1

| (OR) Operator:- Sets each bit to 1 if one of two bits is 1

^ (XOR) Operator:- Sets each bit to 1 if only one of two bits is 1

~ (NOT) Operator:- Inverts all the bits

<< (Zero fill left shift) Operator:- Shift left by pushing zeros in from the right and let the leftmost bits fall off

>> (Signed right shift) Operator:- Shift right by pushing

copies of the leftmost bit in from the left, and let the rightmost bits fall off

Example:-

```
#!/usr/bin/python

a = 60            # 60 = 0011 1100
b = 13            # 13 = 0000 1101
c = 0

c = a & b;        # 12 = 0000 1100
print "Line 1 - Value of c is ", c

c = a | b;        # 61 = 0011 1101
print "Line 2 - Value of c is ", c

c = a ^ b;        # 49 = 0011 0001
print "Line 3 - Value of c is ", c

c = ~a;           # -61 = 1100 0011
print "Line 4 - Value of c is ", c

c = a << 2;       # 240 = 1111 0000
```

```
print "Line 5 - Value of c is ", c

c = a >> 2;       # 15 = 0000 1111
print "Line 6 - Value of c is ", c
```

Output:-

```
Line 1 - Value of c is 12
Line 2 - Value of c is 61
Line 3 - Value of c is 49
Line 4 - Value of c is -61
Line 5 - Value of c is 240
Line 6 - Value of c is 15
```