PYTHON BITWISE OPERATORS EXAMPLE

There are following Bitwise operators supported by Python language

Operator	Description	Example
& Binary AND	Operator copies a bit to the result if it exists in both operands	a & b means00001100
| Binary OR	It copies a bit if it exists in either operand.	$a \mid b$ = 61 means00111101
^ Binary XOR	It copies the bit if it is set in one operand but not both.	a^b = 49 means00110001
~ Binary Ones Complement	It is unary and has the effect of 'flipping' bits.	a = -61 (means 1100 0011 in 2's complement form due to a signed binary number.
<< Binary Left Shift	The left operands value is moved left by the number of bits specified by the right operand.	a << = 240 means11110000
>> Binary Right Shift	The left operands value is moved right by the number of bits specified by the right operand.	a >> = 15 means00001111

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
```

When you execute the above program it produces the following result –

```
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
```