The term operator refers to a symbol (or sometimes a phrase of alphabets) which is predefined to perform a certain process such as addition, comparison etc. Each symbol requires one or more objects for the process to be performed. The objects are called operands and symbol itself is called operator. Most of the operators are binary in nature, in the sense they require two operands. Unary operator requires only one operand.

Arithmetic Operator

Everybody is familiar with arithmetic operators performing addition, subtraction, multiplication and division. Python has additionally modulus exponent and floor operators.

Operator	Purpose	Description
+	Addition	Adds operands on either side of the operator.
-	Subtraction	Subtracts right hand operand from operand on left.
*	Multiplication	returns Multiplication of values on either side of the operator.
/	Division	left operand acts as numerator and right operand denominator for division
%	Modulus	returns remainder of division of left hand operand by right.
**	Exponent	Calculates value of operand raised to right. a**3 is a raised to 3 i.e. a*a*a
//	Floor Division	The division of operands where the result is the quotient in which the digits after the decimal point are removed. But division is negative, the result is floored, i.e., rounded away from zero (towards negative infinity).

In place Assignment Operator

These operators allow any arithmetic operation and assignment in one step. Result of corresponding arithmetic operation of two operands is assigned back to left operand.

symbol	purpose	Description
=	assignment	Assigns values from right side operands to left side operand
+=	Add AND assign	adds right operand to the left and assign result to left operand
-=	Subtract AND assign	subtracts right operand from left and assign the result to left operand
*=	Multiply AND assign	multiplies right operand with left and assign result to left operand
/=	Divide AND assign	divides left operand with right and assign result to left operand
%=	Modulus AND assign	assigns modulus of two operands and assigns result to left operand
**=	Exponent AND assign	Performs exponentiation of left operand by right and assign result to the left operand
//=	Floor Division and assign	It performs floor division on operators and assign value to the left operand

List:

Lists are used to store multiple items in a single variable. Lists are one of 4 built-in data types in Python used to store collections of data, the other 3 are Tuple, Set, and Dictionary, all with different qualities and usage. Lists are created using square brackets:

```
Mylist= [‘apple’, ‘10’, ‘cost’, ‘120.5’]
print(Mylist)
```

List items are indexed, the first item has index [0], the second item has index [1] etc.

Dictionary:

Dictionaries are used to store data values in key: value pairs. A dictionary is a collection which is ordered (in latest python version), changeable and does not allow duplicates. Dictionaries are written with curly brackets, and have keys and values:

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
```

```
}
print(thisdict)
```

Logical Operator:

In order to compare two objects for equality or to decide whether one is greater than other etc. the logical comparison operators are used. Primarily used with numeric objects, they can very well be used with other Python objects such as string, list or tuple.

Symbol	purpose	Description
==	equals	returns true if both operands are equal false otherwise
!=	not equal to	returns true if both operands are not equal false otherwise
>	greater than	returns true if left operand is greater than right operand, otherwise false
<	less than	returns true if left operand is less than right operand, otherwise false
>=	greater than or equal to	returns true if left operand is greater or equal to right operand, otherwise false
<=	less than or equal to	returns true if left operand is less than or equal to right operand, otherwise false

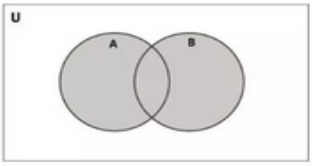
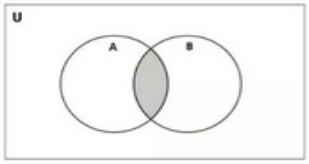
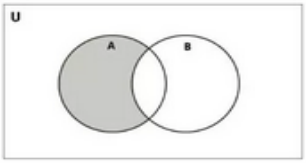
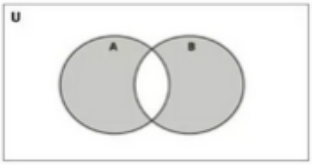
Sequence Operator:

This category of operators is common to all sequence data type i.e. string, list and tuple. All of them use zero based index to access items in them. Hence operators for indexing and slicing have been commonly defined.

symbol	purpose	Description
+	Concatenation	Appends second sequence to first
*	Repetition	concatenates multiple copies of the same sequence
[]	Slice	Gives the item at given index
[:]	Range Slice	fetches item in range specified by two index operands separated by : symbol. If first operand is omitted, range starts at zero index If second operand is omitted, range goes upto end of sequence
in	Membership	Returns true if a item exists in the given sequence
not in	Membership	Returns true if a item does not exist in the given sequence

Set Operator:

These operators are specifically designed for performing operations on set data type as defined in set theory of Mathematics.

purpose	description
Union	Union of two sets is a set of all elements in both. 
Intersection	Intersection of two sets is a set containing elements common to both 
Difference	Difference of two sets results in a set containing elements only in first but not in second set. 
Symmetric difference	Result of Symmetric difference is a set consisting of elements in both sets excluding common element 

Problems Based on Operator:

1. a = 30

   b = 15

   print(a <= b)

   print(a >= b)

   Solution:

   False
   True

2. a = 50

   b = 35

   print(a>b)

   print(a<b)

   Solution:

   True
   False

3. Print the output
   a= 2
   b= 1
   print(a*b**a)
   print(a**b*a)

   Solution:
   2
   4

4. Print the output
   a=c= 6
   b=4
   print(a+b/a)
   print(a*b-c)

   Solution:
   6.6667
   18

5. Print the outputs:

```
a = 4
b = 3
print(+a)
print(-b)
print(a + b)
print(a - b)
print(a * b)
print(a / b)
```

Solution

```
4
-3
7
1
12
1.3333333333333333
```

6. Predict the outputs:

```
a=9
b=3
print(a// b)
print(a// -b)
print(-a // b)
print(-a // -b)
```

Solution:

```
3
-3
-3
3
```

7. Predict the output

```
x = 15
y = 25
z = 6
print(not x > 10 and y > 8 or z < 10)
```

print(x < 10 and not y > 6 or z < 5)

Solution:

True
False

```
my_string = 'Hello'
print(my_string)

my_string = "Hello"
print(my_string)

my_string = "'Hello"'
print(my_string)

my_string = """Hello, welcome to
        the world of Python"""
print(my_string)
print(my_string[5])
print(my_string[1:7])

print(my_string[2:5])
print(my_string[::-1])
```

Solution:

Hello
Hello
Hello
Hello, welcome to
the world of Python
,
ello,
llo
nohtyP fo dlrow eht
ot emoclew ,olleH

8. A person 'X' gets the monthly income by getting basic salary of N Rup, along with it he is getting HRA i.e., 20% of basic, 10% special allowance, 10% conveyance allowance, 15% of medical allowance, and 10% of employee provident fund, but he has tax deduction of 30% on the monthly income. Make use of operator to find out the total salary, also write a program.

Solution:

```
N= eval(input())
HRA= (N*20)/100
S_A= (N*10)/100
C= (N*10)/100
Med= (N*15)/100
PF=(N*10)/100
Monthly_income= N+HRA+S_A+C+Med+PF
print(Monthly_income)
Overall_income= (Monthly_income)- (Monthly_income*30)/100
print(Overall_income)
```

9. Wind chill factor is the felt air temperature on exposed skin due to wind. The wind chill temperature is always lower than the air temperature. Write a program to receive values of temperature and wind velocity and calculate wind chill factor, Wind chilling factor is calculated as per the following formula:

$$Wcf=35.74 + 0.6215t + (0.4275t – 35.75) * V^{(0.16)}$$

Solution:

```
t=float(input())

V=float(input())

Wcf=35.74 + 0.6215*t + (0.4275*t - 35.75) * V**(0.16)

print('%.2f'%Wcf)
```

10. As per the currency system, there are notes/coins of six denominations, Rs. 1, 2, 5, 10, 50, and 100 are given. Suppose the sum of Rs. N is calculated, provide a solution also write a program to combine to give Rs. N with least number of coins.

Solution:

```
n=int(input())
n_100 = n//100
n = n%100
n_50 = n//50
n = n%50
n_10 = n//10
n = n%10
n_5 = n//5
n = n%5
```

```
n_2 = n//2
n = n%2
n_1 = n//1
n = n%1
print("Number of 100 rup note is {}, Number of 50 rup note is {}, Number of 10 rup note is {}, Number of 5 rup note is {}, Number of 2 rup note is {}, and Number of 1 rup note is {}".format(n_100,n_50,n_10,n_5,n_2,n_1))
```

11. Make use of left shift operator and print the power of 2 upto 10 digits.

```
var =  2
var =  4
var =  8
var =  16
var =  32
var =  64
var =  128
var =  256
var =  512
var =  1024
var =  2048
```

Solution:

```
var = 2;

print("var = ", var<<0)
print("var = ", var<<1)
print("var = ", var<<2)
print("var = ", var<<3)
print("var = ", var<<4)
print("var = ", var<<5)
print("var = ", var<<6)
print("var = ", var<<7)
print("var = ", var<<8)
print("var = ", var<<9)
print("var = ", var<<10)
```

12. An expression is given, find out the value of d whether it will be true of false. Take a= 10, b=12, and c=0 and

```
d = (a !=6 and b>5 or a == 9 or b<3 + (a<10)+ (a>5 and c))
```

Solution:

((a!=6) and(b>5) or (a==9) or (((b<3)+(a<10))+((a>5) and c))))

Program:

```
d=(a !=6 and b>5 or a == 9 or b<3 + (a<10)+ (a>5 and c))

print(d)
```

13. A train 280 m long crosses the bridge 170 m in 22.5 seconds. Find the speed of the train in km/hr, also write a program.

Solution:
Length of the train = 280 m
Length of the bridge = 170 m
Therefore, length of the train + length of the bridge = 280 m + 170 m = 450 m
Time taken by the train to cross the bridge = 22.5 sec
Therefore, speed of train = 450 m/22.5 sec = 20 m/sec
To convert the speed from m/sec to km/hr, multiply by 18/5
Therefore, speed of the train = 20 × 18/5 km/hr = 72 km/hr

Program:

```
t_l=float(input())

b_l=float(input())

time_taken= float(input())

total_lenght= t_l+b_l

speed= total_lenght/time_taken

speed_kmh=18/5*speed

print("speed of the train is {}km/hr" .format(speed_kmh))
```

14. Take two number 25 and 14, convert it into binary number format, now perform the XOR using bitwise operator.

Solution:

N1=29	0	0	0	1	1	1	0	1
N2= 15	0	0	0	0	1	1	1	1
Sum= 44	0	0	0	1	0	0	1	0

```
n1= 29
n2= 15
```

```
print(n1^n2)
```