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

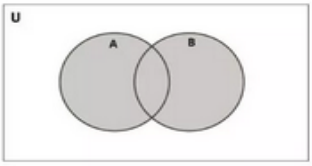
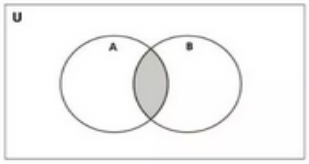
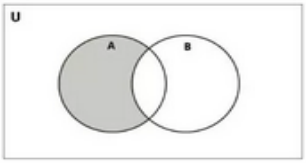
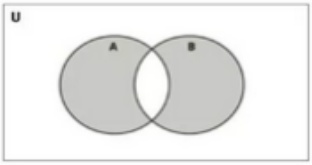
Sequence Operator:

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 = 50

   b = 35

   print(a>b)

   print(a<b)

2. Predict the outputs:
   a=5

   b=2

   print(a % b)

   print(a ** b)

   print(10 / 4)

3. Predict the outputs:
   a= 3
   b= 1
   print(a*b**a)
   print(a**b*a)

4. Predict the outputs:
   a= 2
   b= 4
   print(a-b*a)
   print(a*b/a)

5. Predict the output
   x = 15
   y = 25
   print(x > 10 or y < 8)
   print(x > 10 or y > 8)

   print(x < 10 or y > 8)

6. Predict the output
   x = 24
   y = 20
   list = [10, 20, 30, 40, 50];

print(x **not in** list)
print(y **not in** list)
print(x **in** list)
print(y **in** list)

7. Predict the outputs:
   assorted_list = [True, False, 1, 1.1, 1+2j, "Learn", "b","Python"]

   first_element = assorted_list[0]

   print(first_element)

   first_element = assorted_list[3]

   print(first_element)

   first_element = assorted_list[3]

   print(first_element)

   print(assorted_list[5])

   print(assorted_list)

8. Predict the outputs:

sample_dict = {"key":"value", "jan":31, "feb":28, "mar":31}

print(type(sample_dict))

print(sample_dict)

9. Enter a N digit number and write a program to calculate the sum of its digit.
   a. N= 3
   b. N=5
   c. N=7

10. A paper of size A0 is having the dimension of 1189 mm * 841 mm. Each subsequent size An is defined as cut An-1 in half parallel to its larger sides. Write a program to calculate and print paper sizes A0, A1, A2,..., A8.

11. If a five-digit number is input through the keyboard, write a program to print a new number by adding one to each of its digits. For example (input number : 23679 then output: 34790)

12. Total selling price of 5 item and profit earned on these items is given. Use proper operator to calculate the cost of one item.

13. 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) or 5)

14. A train 340 m long is running at a speed of 45 km/hr. what time will it take to cross a 160 m long tunnel?

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