**Task 1: Arithmetic Operators**

1. Create two variables a and b with numeric values.
2. Calculate the sum, difference, product, and quotient of a and b.
3. Print the results.
4. #Arithmeic operators  
   a=8  
   b=10  
   print(a+b) 🡪 18  
   print(a-b) 🡪-2  
   print(a\*b) 🡪80  
   c=a//b  
   print(c)🡪 0  
   d=b//a  
   print(d)🡪 1

**Task 2: Comparison Operators**

1. Compare the values of a and b using the following comparison operators: <, >, <=, >=, ==, and !=.
2. Print the results of each comparison.

a=20  
b=10  
print(a<b)🡪False  
print(a>b)🡪 True  
print(a==b)🡪 False  
print(a>=b)🡪 True  
print(a<=b)🡪 False  
print(a!=b)🡪 True

**Task 3: Logical Operators**

1. Create two boolean variables, x and y.
2. Use logical operators (and, or, not) to perform various logical operations on x and y.
3. Print the results.

x=True  
y=False  
print(x and y)🡪 False   
print(x or y)🡪 True  
print( not x)🡪 False  
print( not y)🡪 True

**Task 4: Assignment Operators**

1. Create a variable total and initialize it to 10.
2. Use assignment operators (+=, -=, \*=, /=) to update the value of total.
3. Print the final value of total.

total = 10  
total += 5  
print(total)🡪 15  
total -= 5  
print(total)🡪 10  
total \*= 5  
print(total)🡪 50  
total /= 5  
print(total)🡪 10.0

**Task 5: Bitwise Operators (Optional)**

1. If you are comfortable with bitwise operators, perform some bitwise operations on integer values and print the results. If not, you can skip this task.

a=6  
b=3  
print('a & b', a & b)🡪2  
print(a | b)🡪 7   
print( a ^ b)🡪 5  
print( ~ a)🡪 -7  
print(a<<1)🡪 12  
print(b<<2)🡪 12

**Task 6: Identity and Membership Operators**

1. Create a list my\_list containing a few elements.
2. Use identity operators (is and is not) to check if two variables are the same object.
3. Use membership operators (in and not in) to check if an element is present in my\_list.
4. Print the results.

my\_list=[10,3,20,30,50]  
list\_copy = my\_list  
print(my\_list is list\_copy)🡪 True  
print(list\_copy is not my\_list)🡪 False  
  
print(10 in my\_list)🡪 True  
print(44 in my\_list)🡪 False  
print(3 not in my\_list)🡪 False

List Operations:

lst=[10,'uday',3.0,True]  
print('Length\_lst',len(lst)) # o/p --> Length\_lst 4  
  
print('Acessing',lst[0]) # o/p --> Acessing 10

lst[1]='puropale'  
print('modified\_lst',lst) # o/p --> modified\_lst [10,'puropale', 3.0, True]

lst.append(33)  
print('appended\_lst',lst) # o/p--> appended\_lst [10, 'puropale', 3.0, True, 33]  
  
lst.insert(2,'kishore')  
print('new list',lst) # o/p--> new list [10, 'puropale', 'kishore', 3.0, True, 33]   
   
lst.remove(3.0)  
print(lst) # o/p--> [10, 'puropale', 'kishore', True, 33]  
  
sliced\_lst=lst[1:3]  
print('sliced\_lst',sliced\_lst) #o/p--> sliced\_lst ['puropale', 'kishore']  
  
print('membership\_lst', 2 in lst) #o/p--> membership\_lst False

my\_list=[22,43,65,12,99]   
my\_list.sort()  
print('sorted\_lst',my\_list) #o/p--> sorted\_lst [12, 22, 43, 65, 99]  
my\_list.reverse()  
print('reversed\_lst',my\_list) #o/p --> reversed\_lst [99, 65, 43, 22, 12]

for val in my\_list:  
 print('Iterate over\_list', val) #o/p --> Iterate over\_list 99

Iterate over\_list 65

Iterate over\_list 43

Iterate over\_list 22

Iterate over\_list 12

Tuple operations:

my\_tuple=(11,22,33,44,55)  
print('length', len(my\_tuple)) 🡪 o/p length 5  
  
print('my\_tuple',my\_tuple[0]) 🡪 my\_tuple 11

sliced\_tuple = my\_tuple[1:3]  
print("Sliced tuple:", sliced\_tuple) 🡪 Sliced tuple: (22, 33)  
  
print('membership',23 in my\_tuple) 🡪 membership False  
print('membership',33 not in my\_tuple) 🡪 membership False  
  
x= sorted(my\_tuple)  
print("Sorted tuple (as list):",x) 🡪 Sorted tuple (as list): [11, 22, 33, 44, 55]  
  
y= my\_tuple[::-1]  
print('reversed tuple', y) 🡪 reversed tuple (55, 44, 33, 22, 11)

for val in my\_tuple:  
 print(val) 🡪 11

22

33

44

55