**Task 1: Arithmetic Operators**

1. Create two variables a and b with numeric values.

a=10  
b=20  
print(a) #10  
print(b) #20

1. Calculate the sum, difference, product, and quotient of a and b.

a=20  
b=10  
print("Sum of two numbers :", a+b)  
print("Subtraction of two numbers :",a-b)  
print('Multiplication of two numbers :',a\*b)  
print('Division of two numbers :',a/b)

1. Print the results.

Sum of two numbers : 30

Subtraction of two numbers : 10

Multiplication of two numbers : 200

Division of two numbers : 2.0

**Task 2: Comparison Operators**

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

a=5  
b=10  
print(a," less than ",b," :",a<b)  
print(a," greater than ",b," :",a>b)  
print(a," less than or equal ",b," :",a<=b)  
print(a," greater than or equal ",b," :",a>=b)  
print(a," equals ",b," :",a==b)  
print(a," not equals ",b," :",a!=b)

1. Print the results of each comparison.

5 less than 10 : True

5 greater than 10 : False

5 less than or equal 10 : True

5 greater than or equal 10 : False

5 equals 10 : False

5 not equals 10 : True

**Task 3: Logical Operators**

1. Create two boolean variables, x and y.

a=True  
b=False  
print(a) # True  
print(b) # False

1. Use logical operators (and, or, not) to perform various logical operations on x and y.

a=True  
b=False  
print('Logical and :',a and b)  
print('Logical or :',a or b)  
print('Logical not :',not b)

1. Print the results.

Logical and : False

Logical or : True

Logical not : True

**Task 4: Assignment Operators**

1. Create a variable total and initialize it to 10.

a=10

1. Use assignment operators (+=, -=, \*=, /=) to update the value of total.

a=10  
a+=2  
print("Addition :",a)  
a-=2  
print("Subtraction :",a)  
a\*=2  
print("Multiply :",a)  
a/=2  
print('Division :',a)

1. Print the final value of total.

Addition : 12

Subtraction : 8

Multiply : 20

Division : 5.0

The final total value of a : 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=5  
b=3  
print('Bitwise and :',a&b)  
print('Bitwise or :',a|b)  
print('Bitwise not :',~b)  
print('Bitwise xor :',a^b)  
print('Bitwise Left shift :',a<<2)  
print('Bitwise right shift :',b>>1)

OUTPUT:

Bitwise and : 1

Bitwise or : 7

Bitwise not : -4

Bitwise xor : 6

Bitwise Left shift : 20

Bitwise right shift : 1

**Task 6: Identity and Membership Operators**

1. Create a list my\_list containing a few elements.

my\_list=[10,20,30,10]  
print(my\_list) # [10, 20, 30, 10]

1. Use identity operators (is and is not) to check if two variables are the same object.

my\_list=[10,20,30,40]  
list1=my\_list  
print(list1 is my\_list) # True  
print(list1 is not my\_list) # False

1. Use membership operators (in and not in) to check if an element is present in my\_list.

my\_list=[10,20,30,10]  
print(10 in my\_list) # True   
print(50 not in my\_list) #True  
print(20 not in my\_list) #False   
print(40 in my\_list) #False

1. Print the results.

Results of Identity Operator:

True

False

Results of Membership Operator:

True

True

False

False

**Lists:**

Creating a list.

l=[10,20,30,40]  
print(l) # [10, 20, 30, 40]

Length of the list:

l=[10,20,30,40]  
print("Length of the list :",len(l)) # Length of the list : 4

Accessing elements:

l=[10,20,30,40]  
print(l[1]) ---20 # by using +ve index  
print(l[-2]) ---30 # by using -ve index

Adding elements to the list:

l=[10,20,30,40]  
l.append(15) # adding elements  
print('Adding element to the list :',l)  
l.insert(2,50) #adding at specified position  
print('Adding an element at the specified position :',l)  
l.extend([90,70]) #extending list  
print('Extending the list :',l)

Output:

Adding element to the list : [10, 20, 30, 40, 15]

Adding an element at the specified position : [10, 20, 50, 30, 40, 15]

Extending the list : [10, 20, 50, 30, 40, 15, 90, 70]

Removing the elements from the list:

l=[10,20,30,40]  
l.pop(4) # removing the specified element using position  
print('Removing the specified element using position :',l)  
l.remove(20)  
print('Removing the specified element :',l)  
l.pop()  
print("Removing the element if the position is not specified :",l)

l.clear()

print(l)

Output:

Removing the specified element using position : [10, 20, 50, 30, 15, 90, 70]

Removing the specified element : [10, 50, 30, 15, 90, 70]

Removing the element if the position is not specified : [10, 50, 30, 15, 90]

[]

Slicing:

print("Getting the elements within the range :",l[1:3])  
print("Getting the elements from the start to the given index :",l[:3])  
print("Getting the elements from the given range to the end of the list",l[1:])  
print("Getting all the elements from the list :",l[:])

Output:

Getting the elements within the range : [50, 30]

Getting the elements from the start to the given index : [10, 50, 30]

Getting the elements from the given range to the end of the list [50, 30, 15, 90]

Getting all the elements from the list : [10, 50, 30, 15, 90]

Reverse a list:

l.reverse()  
print('Reverse the list :',l)

Output:

Reverse the list : [90, 15, 30, 50, 10]

Sort the list:

l.sort()  
print('Sorting the list :',l)

Output:

Sorting the list : [10, 15, 30, 50, 90]

Copy of the list:

l1=l.copy()  
print('Copy of the list :',l1)

Output:

Copy of the list : [10, 15, 30, 50, 90]

Count:

l.append(10)  
print('Count of the element :',l.count(10))

Output:

Count of the element : 2

Iteration:

for i in l:  
 print(i)

Output:

10

15

30

50

90

10

Membership:

if 10 in l:  
 print(True)

Output:

True

**Tuple:**

Creating a tuple:

t=(10,20,50,30)

print(t)

Output:

(10,20,50,30)

Length of the tuple:

t=(10,20,50,30)

print(‘Length of the tuple :’,len(t))

Output:

Length of the tuple :4

Accessing elements of the tuple:

t=[10,20,50,30]  
print(l[1]) ---20 # by using +ve index  
print(l[-2]) ---50 # by using -ve index

Slicing:

print("Getting the elements within the range :",t[1:3])  
print("Getting the elements from the start to the given index :",t[:3])  
print("Getting the elements from the given range to the end of the tuple",t[1:])  
print("Getting all the elements from the tuple :",t[:])

Output:

Getting the elements within the range : [20,50]

Getting the elements from the start to the given index : [10, 20, 50]

Getting the elements from the given range to the end of the tuple [20,50, 30]

Getting all the elements from the tuple : [10,20, 50, 30]

Count:

print('Count of the element :',l.count(10))

Output:

Count of the element :1

Iteration:

for i in t:  
 print(i)

Output:

10

20

50

30

Membership:

if 10 in t:  
 print(True)

Output:

True