

# – Task 20 –

Basic and Advanced Data Manipulation with NumPy

Name: Basel Amr Barakat  
Email: [baselamr52@gmail.com](mailto:baselamr52@gmail.com)

# Requirement One

Creating and Manipulating NumPy Arrays.

```
1 import numpy as np
2 # Creating a NumPy arrays from different datatypes (integers, floats, Booleans)
3 int_numpy_array = np.array([1, 2, 3, 4])
4 float_numpy_array = np.array([1.5, 2.5, 3.5, 4.5])
5 bool_numpy_array = np.array([True, False, True, False])
6
7 # Performing basic arithmetic operations (addition, subtraction, multiplication)
8 # Addition
9 summition_array = int_numpy_array + float_numpy_array
10 # Subtraction
11 subtraction_array = int_numpy_array - float_numpy_array
12 # Multiplication
13 multiplication_array = int_numpy_array * float_numpy_array
14 # Division
15 division_array = int_numpy_array / float_numpy_array
16
17 # Accessing elemnts from array (indexing)
18 element_one_int = int_numpy_array[1] #-----> Accessing the second element in the int_numpy_array
19 element_one_float = float_numpy_array[0] #-----> Accessing the first element in the float_numpy_array
20 element_one_bool = bool_numpy_array[2] #-----> Accessing the third element in the bool_numpy_array
21
22 # Displaying the results
23 print("int_numpy_array:", int_numpy_array)
24 print("float_numpy_array:", float_numpy_array)
25 print("bool_numpy_array:", bool_numpy_array)
26 print("summition_array:", summition_array)
27 print("subtraction_array:", subtraction_array)
28 print("multiplication_array:", multiplication_array)
29 print("division_array:", division_array)
30 print("element_one_int:", element_one_int)
31 print("element_one_float:", element_one_float)
32 print("element_one_bool:", element_one_bool)
```

```
int_numpy_array: [1 2 3 4]
float_numpy_array: [1.5 2.5 3.5 4.5]
bool_numpy_array: [ True False  True False]
summition_array: [2.5 4.5 6.5 8.5]
subtraction_array: [-0.5 -0.5 -0.5 -0.5]
multiplication_array: [ 1.5  5.  10.5 18. ]
division_array: [0.66666667 0.8          0.85714286 0.88888889]
element_one_int: 2
element_one_float: 1.5
element_one_bool: True
```

# Requirement Two

Advanced Array Manipulation.

```

1 # Reshaping arrays
2 reshaped_array_34 = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]).reshape(3, 4) # Reshaping into a 3x4 array
3 reshaped_array_43 = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]).reshape(4, 3) # Reshaping into a 4x3 array
4 reshaped_array_2 = reshaped_array_34.reshape(2, -1) # Reshaping into a 2x? array (automatically determined)
5
6 # Stacking arrays (Vertical Stack and Horizontal Stack)
7 vstack_array = np.vstack([reshaped_array_34, reshaped_array_34])
8 hstack_array = np.hstack([reshaped_array_43, reshaped_array_43])
9
10 # Splitting arrays (Split into 3 equal parts)
11 split_array = np.split(reshaped_array_34, 3) # Splitting reshaped_array into 3 parts
12
13 # Boolean indexing (Filter elements greater than 5)
14 filtered_array = reshaped_array_34[reshaped_array_34 > 5] # Filtering elements greater than 5 and less than 10
15
16 # Displaying results
17 print("\nReshaped Array (3x4):")
18 print(reshaped_array_34)
19 print("\nReshaped Array (4x3):")
20 print(reshaped_array_43)
21 print("\nReshaped Array (2x?):")
22 print(reshaped_array_2)
23
24 print("\nVertically Stacked Array:")
25 print(vstack_array)
26 print("\nHorizontally Stacked Array:")
27 print(hstack_array)
28 print("\nSplit Array (3 parts):")
29 print(split_array)
30 print("\nFiltered Array (elements greater than 5):")
31 print(filtered_array)

```

```

Reshaped Array (3x4):
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]

Reshaped Array (4x3):
[[ 1  2  3]
 [ 4  5  6]
 [ 7  8  9]
 [10 11 12]]

Reshaped Array (2x?):
[[ 1  2  3  4  5  6]
 [ 7  8  9 10 11 12]]

Vertically Stacked Array:
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]
 [ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]

Horizontally Stacked Array:
[[ 1  2  3  1  2  3]
 [ 4  5  6  4  5  6]
 [ 7  8  9  7  8  9]
 [10 11 12 10 11 12]]

Split Array (3 parts):
[array([[1, 2, 3, 4]]), array([[5, 6, 7, 8]]), array([[ 9, 10, 11, 12]])]

Filtered Array (elements greater than 5):
[ 6  7  8  9 10 11 12]

```

# Requirement Three

Mathematical and Statistical Operations..

```
1 # Performing sum, mean, min, max, etc., using NumPy
2 array_for_operations = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
3
4 total_sum = np.sum(array_for_operations) # Sum of elements
5 mean_value = np.mean(array_for_operations) # Mean of elements
6 min_value = np.min(array_for_operations) # Minimum value
7 max_value = np.max(array_for_operations) # Maximum value
8 std_dev = np.std(array_for_operations) # Standard deviation
9 # Another way to calculate STD
10 std_dev_another_way = np.sqrt(np.sum((array_for_operations - np.mean(array_for_operations))**2/len(array_for_operations)))
11
12 # Displaying results
13 print("\nMathematical and Statistical Operations:")
14 print("Total Sum:", total_sum)
15 print("Mean Value:", mean_value)
16 print("Minimum Value:", min_value)
17 print("Maximum Value:", max_value)
18 print("Standard Deviation:", std_dev)
19 print("Standard Deviation (Another Way):", std_dev_another_way)
```



```
Mathematical and Statistical Operations:
Total Sum: 55
Mean Value: 5.5
Minimum Value: 1
Maximum Value: 10
Standard Deviation: 2.8722813232690143
Standard Deviation (Another Way): 2.8722813232690143
```