Classroom Assignment <20>: Working with NumPy Arrays



**Learning Objective:**

Learn how to **create, access, modify, reshape, join, and split** NumPy arrays using a test data scenario.

**Expected Completion Time:**

Best Case: 20 minutes Average Case: 30 minutes

# Assignment Details:

Create a Python script named **numpy\_array\_basics.py** that demonstrates the following:

1. **Creating**
   * Create a 1D NumPy array with execution times of 8 test cases:
   * [10, 15, 20, 25, 30, 35, 40, 45]
2. **Indexing & Shaping**
   * Access the **first**, **last**, and **3rd** element of the array.
   * Print the **shape** of the array.
3. **Slicing**
   * Print execution times of the **first 3 tests**.
   * Print every **alternate test time**.
4. **Iteration**
   * Iterate through the array and print each execution time with a message:  
     "Test X execution time: Y seconds".
5. **Reshaping**
   * Reshape the 1D array (8 elements) into a **2D array of shape (2,4)**.
   * Print the reshaped array.
6. **Joining**
   * Create another NumPy array with 4 more execution times:
   * [50, 55, 60, 65]
   * Join (concatenate) this with the first array to form a longer array.
7. **Splitting**
   * Split the final array into 3 smaller arrays (equal parts if possible).
   * Print each split.

# Requirements:

 Use numpy.array() to create arrays.

 Use indexing (arr[0], arr[-1]) for access.

 Use slicing (arr[0:3], arr[::2]) for subsets.

 Use .shape, .reshape(), np.concatenate(), and np.array\_split().

**Hints to Solve:**

 Reshaping requires compatible dimensions (8 elements → (2,4) works).

 np.concatenate([arr1, arr2]) merges arrays.

 np.array\_split(arr, 3) splits into 3 parts.

# Expected Outcome:



Upon completion, you should be able to:

* Create and manipulate NumPy arrays.
* Use indexing and slicing for data access.
* Reshape arrays into multi-dimensional structures.
* Join and split arrays for flexible data handling.
* Iterate through arrays for reporting.