**Lab 8**

**Submission Guidelines:** Include all code and outputs in a Jupyter Notebook.

**Question 1**

**a)** Create a Numpy array containing the numbers from 1 to 10.

**b)** Reshape the array into a 2x5 matrix.

**c)** Extract the second row of the matrix.

**d)** Calculate the mean of each column in the matrix.

**Question 2**

**a)** Use Numpy's arange function to create an array of numbers from 0 to 20 with a step of 2.

**Question 3**

Given the array arr = np.array([10, 20, 30, 40, 50, 60]):

**a)** Slice the array to get elements from index 2 to 4.

**b)** Reverse the array.

**c)** Assign the value 100 to the last element of the array.

**Question 4**

**a)** Create an array x containing 100 equally spaced numbers between 0 and 2π.

**b)** Compute the sine and cosine of all elements in x and store them in arrays sin\_x and cos\_x.

**c)** Find the product of sin\_x and cos\_x.

**Question 5: Creating and Manipulating Pandas Series**

**a)** Create a Pandas Series s from the list [1, 3, 5, 7, 9] with index labels ['a', 'b', 'c', 'd', 'e'].

**b)** Retrieve the element at index label 'c'.

**c)** Change the value at index label 'd' to 10.

**d)** Add a new element with index label 'f' and value 12 to the Series.

**Question 6**

**a)** Create a DataFrame df from the following dictionary:

data = {

'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eva'],

'Age': [24, 27, 22, 32, 29],

'City': ['New York', 'Los Angeles', 'Chicago', 'Houston', 'Phoenix'],

'Score': [85.5, 90.0, 78.0, 88.5, 95.0]

}

**b)** Display the first two rows of the DataFrame.

**c)** Get summary statistics of the numerical columns.

**d)** Add a new column 'Passed' to the DataFrame where the value is True if 'Score' is greater than or equal to 80, else False.

**Question 7**

Create a DataFrame df from the following dictionary:

data = {

'Name': ['Tom', 'Joseph', 'Krish', 'John', 'Mike'],

'Age': [20, np.nan, 22, np.nan, 25],

'Qualification': ['Bachelor', 'Master', np.nan, 'PhD', 'Bachelor']

}

**a)** Identify the rows with missing values.

**b)** Drop the rows with missing values and display the resulting DataFrame.

**c)** Fill the missing values in 'Age' with the mean age and missing values in 'Qualification' with Bachelor. Show the updated DataFrame.

**Question 8**

Using the DataFrame df from **Question 6**:

**a)** Group the DataFrame by 'City' and calculate the mean 'Age' for each city.

**b)** For each 'Passed' status, count the number of students.

**c)** Group the DataFrame by 'Passed' and calculate the average 'Score'.