

## QUIZ

# NumPy & Pandas

*Python Data Science Series*

### STUDENT INFORMATION

FULL NAME

Aditya Kumar

CLASS / SECTION

ROLL NUMBER

Date: \_\_\_\_\_ 23 Feb 2026 \_\_\_\_\_

Total Marks: \_\_\_\_\_ / 30

SECTION 1

# NumPy Arrays & Operations

*Questions 1–5 · Broadcasting, Indexing & Numerical Computing*

5 Questions × 1 Mark = 5 Marks

What does the following NumPy operation return?

```
np.array([1, 2, 3]) * 3
```

A [3, 6, 9]

B [1, 2, 3, 1, 2, 3, 1, 2, 3]

C [3, 3, 3]

D Error – cannot multiply array by scalar

Answer: \_\_\_A\_\_\_

Which of the following NumPy array shapes are broadcast-compatible?

A (3, 4) and (5, 4)

B (3,) and (3, 3)

C (4, 1) and (1, 4)

D Both B and C

Answer: \_\_\_D\_\_\_

Given arr = np.array([[1,2,3],[4,5,6],[7,8,9]]), what does arr[1:, ::2] return?

- A [[4, 6], [7, 9]]
- B [[4, 5, 6], [7, 8, 9]]
- C [[4, 6], [7, 9]]
- D [[5, 6], [8, 9]]

Answer: \_\_C\_\_

Which NumPy function computes the dot product of two 1-D arrays?

A np.multiply()

B np.dot()

C np.cross()

D np.sum()

Answer: \_\_\_A\_\_\_

What is the output of `np.arange(2, 10, 3)`?

A [2, 5, 8, 11]

B [2, 5, 8]

C [3, 6, 9]

D [2, 4, 6, 8]

Answer: \_B\_\_\_\_\_

## SECTION 2

# Pandas — Series & DataFrames

*Questions 6–12 · Data Loading, Filtering, Sorting & Wrangling*

7 Questions × 1 Mark = 7 Marks + 3 Short Answer × 2 Marks = 6 Marks

Which Pandas method is used to read a CSV file into a DataFrame?

A `pd.read_table()`

B `pd.read_csv()`

C `pd.load_csv()`

D `pd.import_csv()`

Answer: \_\_\_B\_\_\_

Which parameter in pd.read\_excel() specifies which sheet to load?

A tab\_name

B sheet\_index

C sheet\_name

D worksheet

Answer: \_\_\_C\_\_\_

Given df with a column 'Age', which expression filters rows where Age > 25?

A `df[df.Age > 25]`

B `df.filter(Age > 25)`

C `df.where('Age > 25')`

D `df.query['Age > 25']`

Answer: \_\_A\_\_\_\_

Which method sorts a DataFrame by the 'Score' column in descending order?

- A df.sort('Score', desc=True)
- B df.order\_by('Score', ascending=False)
- C df.sort\_values('Score', ascending=False)
- D df.arrange('Score', reverse=True)

Answer: \_\_\_C\_\_\_

What does `df.groupby('Dept')['Salary'].mean()` return?

A

A single mean salary value

B

Mean salary grouped by department

C

Mean salary of the 'Dept' column

D

Error – cannot group then mean

Answer: \_\_\_ B \_\_\_

Which method combines two DataFrames horizontally (column-wise)?

- A pd.concat([df1, df2], axis=0)
- B pd.concat([df1, df2], axis=1)
- C df1.append(df2)
- D pd.merge(df1, df2, how='outer')

Answer: \_\_\_B\_\_\_

What does `df.dropna()` do to a DataFrame?

A Drops all columns with NaN values

B Fills NaN values with 0

C Drops all rows containing NaN values

D Replaces NaN with the column mean

Answer: \_\_\_C\_\_\_

## SHORT ANSWER

**Explain the split-apply-combine strategy used by groupby() in Pandas. Provide a real-world example showing how groupby() with an aggregation function is used.**

Answer: It is widely used in **corporate analytics, dashboards, finance reports, HR analysis, sales reporting**, etc.

It is used the split apply combine method to manipulate the data.

```
import pandas as pd
```

```
data = {  
    "Employee": ["Aditya", "Rahul", "Priya", "Sneha", "Aman", "Neha"],  
    "Department": ["Sales", "HR", "IT", "Finance", "Sales", "IT"],  
    "Score": [95, 88, 92, 75, 98, 85]  
}
```

```
df = pd.DataFrame(data)
```

## SHORT ANSWER

**What is broadcasting in NumPy? State the broadcasting rules and give an example showing how arrays of shapes (3, 1) and (1, 4) would interact.**

---

Answer: Broadcasting is a powerful NumPy mechanism that allows arrays of **different shapes** to perform arithmetic operations **without explicitly reshaping them**.

---

```
import numpy as np

A = np.array([[1],
              [2],
              [3]]) # Shape (3,1)

B = np.array([[10, 20, 30, 40]]) # Shape (1,4)

result = A + B
print(result)
```

---

## SHORT ANSWER

**Describe any TWO common data-wrangling tasks in Pandas. For each task, name the method used and write a one-line code example.**

Answer : Handling Missing Values

Fill or manage missing (NaN) values in a dataset.

---

```
df["Salary"] = df["Salary"].fillna(df["Salary"].mean())
```

Grouping and Aggregation

---

Summarize data based on categories.

groupby() with aggregation (mean(), sum(), etc.)

```
df.groupby("Department")["Score"].mean()
```

---

# END OF QUIZ

*Please review all answers before submitting.*

Section	Topic	Questions	Marks
Section 1	NumPy Arrays & Operations	Q1–Q5	5 Marks
Section 2	Pandas MCQ	Q6–Q12	7 Marks
Section 2	Short Answer	Q13–Q15	6 Marks
TOTAL			<b>30 Marks</b>