1. Numpy Practice Exercises (Arrays, Indexing, Reshaping)

Exercise 1 — Create Arrays

Create:

1. A 1D array from 1–20
2. A 2D array of shape 4×5
3. A 3D array of shape 2×3×3
4. A zero matrix of size 3×3
5. A matrix filled with 7s (5×5)

Exercise 2 — Array Indexing & Slicing

Given:

```
arr = np.arange(1, 26).reshape(5, 5)
```

Perform:

1. Extract the 3rd row
2. Extract the 2nd column
3. Extract a 2×2 block from the bottom-right corner
4. Extract all even numbers
5. Replace the middle element with 999

Exercise 3 — Reshaping

Convert:

1. 1×12 → 3×4
2. 6×6 → 3×12 → 2×18
3. Flatten a 2D matrix to 1D
4. Convert 1D array into:
    - Column vector
    - Row vector

Exercise 4 — Vectorization vs Loops

Given array:

```
arr = np.arange(1, 100001)
```

Tasks:

- Square each element using a **loop**
- Square each element using **vectorization (arr**2)**
- Compare time using `%timeit`
- Print which approach is faster

Exercise 5 — Conditional Selection

Given an array of student marks:

```
marks = np.array([10, 45, 67, 89, 32, 56, 77, 90, 12])
```

Perform:

- Extract all passing marks (> 40)
- Find marks between 50–80
- Replace all marks < 40 with 0
- Count how many students passed

2. Pandas Practice Exercises (Series & DataFrames)

Exercise 6 — Pandas Series

Create a Series for:

- Five product names
- Price list
  Perform:
- Find max, min
- Filter items with price > 500
- Convert all names to uppercase

Exercise 7 — Create a DataFrame

Create a DataFrame of 6 students with:

- Name
- Age
- Gender
- Score

Then:

- Display head & tail
- Get summary statistics
- Sort by Score descending
- Filter students with Score > 80
- Add a new column “Grade” based on score

Exercise 8 — Indexing & Slicing

Given DataFrame:

```
df = pd.DataFrame({
    "Name": ["A", "B", "C", "D", "E"],
    "Age": [20, 21, 22, 23, 24],
    "Score": [88, 76, 90, 66, 95]
})
```

Do:

- Select rows 0 to 2
- Select Score column
- Select Age & Score columns
- Select students aged > 21
- Select rows where Score > 80

3. Data Cleaning Exercises

(*Load → Clean → Analyze*)

Exercise 9 — Clean a Dirty Dataset (Provided Below)

Dirty Dataset (students.csv):

```
Name,Age,Marks,City
ram,20,87,Bangalore
sita,,91,chennai
john,???,abc,mumbai
ram,20,87,Bangalore
meera,21,,Hyderabad
,19,77,Delhi
kiran,22,85,
```

Tasks:

1. Load CSV using pandas
2. Identify missing values
3. Replace missing Age with mean

4. Replace “???” with NaN → then mean
5. Convert Marks to numeric (coerce errors)
6. Replace missing City with “Unknown”
7. Remove duplicates
8. Convert Name → title case
9. Print summary statistics
10. Print the top 3 highest scorers

Exercise 10 — Column Operations

Given DataFrame:

- Add column “Pass/Fail”
- Add Bonus Marks = Marks + 5
- Drop unnecessary columns
- Rename columns (Marks → Score)
- Change datatype of Age to int

4. Guided Hands-On Tasks (Matching Your Session Plan)

Task 1: Load a CSV Dataset

Use Kaggle sample datasets or teacher-provided datasets.

Examples:

- Titanic
- Students result
- Movies dataset
- Sales dataset

Task 2: Perform Slicing, Filtering, Sorting

Examples:

- Filter by age > 30
- Sort by salary descending
- Show only name + city columns
- Get top 10 highest values

Task 3: Calculate Summary Statistics

Use:

```
df.describe()
df.mean()
df.mode()
df.median()
df.nunique()
```

Try:

- Get median salary
- Get unique categories
- Most frequent city

Task 4: Clean a Dirty Dataset

Combine all steps:

- Drop missing values
- Replace NaN
- Remove duplicates
- Normalize text (lowercase, title case, strip spaces)
- Convert string numbers to integers

5. Fun Task — Build Your Own Mini Netflix Dataset

Students create their own dataset (10–20 rows) with columns like:

- Title
- Year
- Genre
- IMDb Rating
- Duration
- Type (Movie/Series)
- Actor
- Country

Fun Task Part A: Create Dataset

Students manually create a CSV:

```
Title,Year,Genre,Rating,Type,Actor
Inception,2010,Sci-Fi,8.8,Movie,Leonardo DiCaprio
Interstellar,2014,Sci-Fi,8.6,Movie,Matthew McConaughey
...
```

Fun Task Part B: Analyze Dataset

Using pandas:

- Top-rated movie
- Movies after 2015
- All Sci-Fi movies
- Average rating
- Most common genre
- Longest movie
- Movies grouped by Type

Fun Task Part C: Visualizations (optional)

- Bar plot of rating by genre
- Count of movies per year
- Histogram of IMDb ratings

6. BONUS Challenges (For Online Session)

Challenge 1 — Detect Duplicate Movies

Find duplicate titles in Netflix dataset.

Challenge 2 — Replace Bad Ratings

If Rating < 3:

```
df["Rating"] = df["Rating"].replace(value < 3, np.nan)
```

Challenge 3 — Vectorize Duration

Convert:

- “120 min” → 120
- “45 min” → 45