Pandas Assignment – Series & DataFrame

Part A: Theory (Short Answer)

1. What is the difference between a **Series** and a **DataFrame** in Pandas? Give an example.
2. Explain the difference between `loc[]` and `iloc[]` with examples.
3. What are the main use cases of Pandas in real-world applications?
4. List at least 5 important methods of a Series and 5 of a DataFrame with their purpose.
5. What is the difference between `df.info()` and `df.describe()`?

Part B: Hands-On Questions

Q1. Series Creation & Operations

1. Create a Pandas Series from a list `[10, 20, 30, 40, 50]` with index `['a','b','c','d','e']`.
2. Perform:
   - Sum, Mean, Standard Deviation
   - Multiply each element by 2 (vectorized operation)
   - Replace all values greater than 30 with `100`.

Q2. DataFrame Creation & Basic Exploration

Create a DataFrame from the following dictionary:

```
data = {
    'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eva'],
    'Age': [25, 30, 35, 40, 28],
    'Salary': [50000, 60000, 75000, 80000, 62000],
    'City': ['Delhi', 'Mumbai', 'Delhi', 'Chennai', 'Mumbai']
}
```

1.
2. Perform:
   - Display **first 3 rows**
   - Show column names and shape of the DataFrame
   - Access only the `Name` and `Salary` columns
   - Select the rows where `City` = "Delhi"

Q3. Indexing & Selection

1. From the DataFrame in Q2:
   - Select the **2nd row, 3rd column** value using `iloc`
   - Select the **Salary of Bob** using `loc`
   - Slice the DataFrame to show rows 2–4 and columns `Name` & `Salary`

Q4. Data Cleaning

Create a DataFrame with some missing values:

```
df = pd.DataFrame({
    'ID': [1, 2, 3, 4, 5],
    'Marks': [85, np.nan, 78, np.nan, 90],
    'Subject': ['Math', 'Science', 'English', 'History', 'Math']
})
```

1.
2. Perform:

   - Fill missing values in `Marks` with the **mean**
   - Drop rows with missing values
   - Show unique subjects and count their frequency

Q5. Sorting & Aggregation

Using the DataFrame from Q2:

1. Sort the DataFrame by `Salary` in descending order.
2. Find the **average Salary** of all employees.
3. Count how many employees are in each `City`.

Q6. File Handling

1. Save the DataFrame from Q2 into a CSV file named **employee.csv**.
2. Read the CSV back into Pandas and display only the `Name` and `City` columns.
3. Export the DataFrame into **Excel format** (`employee.xlsx`).

Part C: Mini Project (Open-Ended)

- Download or create a CSV file of your choice (at least 10 rows and 5 columns, with a mix of numerical, categorical, and missing values).
- Perform the following:
    1. Read the CSV into a DataFrame
    2. Show top 5 rows, column names, and shape
    3. Handle missing values (fill/drop)
    4. Sort data by one numeric column
    5. Apply at least 3 aggregate functions (`sum`, `mean`, `max`, etc.)
    6. Save the cleaned DataFrame back to a new CSV file