**Section A: Data Creation and Importing (7 Questions)**

1. import pandas as pd

data = {

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

'Age': [25, 30, 35],

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

}

df = pd.DataFrame(data)

print(df)

-Output-

Name Age City

0 Alice 25 New York

1 Bob 30 Los Angeles

2 Charlie 35 Chicago

1. import pandas as pd

df = pd.read\_csv('data.csv')

print(df.head())

1. import pandas as pd

df = pd.read\_excel('data.xlsx')

print(df.head())

-Output-

df = pd.read\_excel('data.xlsx', sheet\_name='Sheet1')

1. import pandas as pd

df = pd.read\_json('data.json')

print(df.head())

1. import pandas as pd

tables = pd.read\_html('data.html')

df = tables[0]

print(df.head())

1. import pandas as pd

data = {

'Product': ['Laptop', 'Phone', 'Tablet'],

'Price': [800, 500, 300]

}

df = pd.DataFrame(data)

print(df)

-Output-

Product Price

0 Laptop 800

1 Phone 500

2 Tablet 300

1. df.head()

df.head(10)

**Section B: Data Inspection (7 Questions)**

1. df.head(10)
2. df.tail(3)
3. df.info()
4. df.describe()

df.mean()

df.median()

df.std()

df['column\_name'].mean()

1. df.dtypes
2. df.shape
3. The df.info() method is a handy tool in pandas for getting a concise summary of a DataFrame's structure. It provides the following details:
4. **Number of Entries:** Displays the total number of rows in the DataFrame and the range of the index.
5. **Column Names and Count:** Lists all column names and their total count.
6. **Non-Null Count:** Shows how many non-null (non-missing) values exist in each column.
7. **Data Types:** Displays the data type (dtype) of each column (e.g., int64, float64, object).
8. **Memory Usage:** Indicates the approximate memory consumed by the DataFrame.

df.info()

-Output-

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 100 entries, 0 to 99

Data columns (total 3 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Name 100 non-null object

1 Age 95 non-null int64

2 City 100 non-null object

dtypes: int64(1), object(2)

memory usage: 2.5 KB

**Section C: Indexing and Selecting Data (8 Questions)**

1. df.set\_index('ID', inplace=True)

df\_new = df.set\_index('ID')

1. df.reset\_index(inplace=True)

df.reset\_index(drop=True, inplace=True)

1. df.iloc[2]

df.iloc[[2]]

1. df.loc[df['Age'] > 30]
2. df.query('Salary > 50000')
3. df.sort\_values(by='Price', ascending=True, inplace=True)
4. top\_rows = df.nlargest(3, 'Marks')
5. bottom\_rows = df.nsmallest(2, 'Age')

**Section D: Data Cleaning (8 Questions)**

1. df.isnull()

df.isnull().sum()

1. df.dropna(inplace=True)

df\_cleaned = df.dropna()

1. df['Salary'].fillna(df['Salary'].mean(), inplace=True)

df\_new = df['Salary'].fillna(df['Salary'].mean())

1. df.drop\_duplicates(inplace=True)

df\_unique = df.drop\_duplicates()

1. df['Gender'].replace('M', 'Male', inplace=True)

df\_updated = df['Gender'].replace('M', 'Male')

1. df['Age'] = df['Age'].astype(int)
2. df.dropna(subset=['Age'], inplace=True)

df\_cleaned = df.dropna(subset=['Age'])

1. df.fillna(method='ffill', inplace=True)

df\_filled = df.fillna(method='ffill')