Question 1: DataFrame Aggregation (5 Marks)

```
import pandas as pd
# 1. Create DataFrame
data = {
    "Salesperson": ["John", "Amy", "John", "Amy", "Tom"],
    "Sales": [200, 300, 150, 400, 250]
}
df = pd.DataFrame(data)

# 2. Group by Salesperson and calculate total sales
result = df.groupby("Salesperson").sum()
print(result)
```

Question 2: Data Cleaning (5 Marks)

```
import pandas as pd
import numpy as np

# 1. Create DataFrame with missing values
data = {
    "Name": ["John", "Amy", "Tom"],
    "Age": [28, np.nan, 35],
    "Salary": [50000, 70000, np.nan]
}
df = pd.DataFrame(data)

# 2. Fill missing values with column mean
df["Age"].fillna(df["Age"].mean(), inplace=True)
df["Salary"].fillna(df["Salary"].mean(), inplace=True)

print(df)
```

Question 3: Data Correlation (5 Marks)

```
import pandas as pd

# 1. Create DataFrame
data = {
    "Math Score": [85, 78, 92, 67, 88],
    "Science Score": [80, 72, 89, 65, 85]
}
df = pd.DataFrame(data)

# 2. Find correlation
correlation = df.corr()
print(correlation)
```

Question 4: Covariance (5 Marks)

```
import pandas as pd

# 1. Create DataFrame
data = {
    "Height": [170, 160, 180, 175],
    "Weight": [65, 60, 80, 75]
}
df = pd.DataFrame(data)

# 2. Compute covariance
covariance = df.cov()
print(covariance)
```

Question 5: Sorting Data (5 Marks)

```
import pandas as pd
# 1. Create DataFrame
data = {
    "Name": ["John", "Amy", "Tom", "Lisa"],
    "Age": [28, 24, 30, 22]
}….df = pd.DataFrame(data)
# 2. Sort by age in descending order
sorted_df = df.sort_values("Age", ascending=False)…print(sorted_df)
```

Question 6: Conditional Selection (5 Marks)

```
import pandas as pd

# 1. Create DataFrame
data = {
    "Employee": ["John", "Amy", "Tom", "Lisa"],
    "Salary": [55000, 40000, 60000, 45000]
}
df = pd.DataFrame(data)

# 2. Filter rows where salary is greater than 50,000
high_salary = df[df["Salary"] > 50000]
print(high_salary)
==================================================================================
```

Question 7: Data Transformation (5 Marks)

```
import pandas as pd

# 1. Create DataFrame
data = {
    "Product": ["A", "B", "C"],
    "Price": ["$100", "$200", "$300"]
}
df = pd.DataFrame(data)

# 2. Convert Price to numeric by removing "$"
df["Price"] = df["Price"].str.replace("$", "").astype(float)
print(df)
```

Question 8: Handling Duplicates (5 Marks)

```
import pandas as pd

# 1. Create DataFrame with duplicate rows
data = {
    "Name": ["John", "Amy", "Tom", "John", "Tom"],
    "Age": [28, 24, 30, 28, 30]
}
df = pd.DataFrame(data)

# 2. Remove duplicate rows
df_no_duplicates = df.drop_duplicates()
print(df_no_duplicates)
```

Question 9: Sampling Data (5 Marks)

```
import pandas as pd

# 1. Create DataFrame with 10 rows
data = {
    "Name": ["John", "Amy", "Tom", "Lisa", "Bob", "Kate", "Mark", "Jim", "Sam", "Rick"],
    "Age": [28, 24, 30, 22, 26, 32, 40, 36, 31, 29]
}
df = pd.DataFrame(data)

# 2. Randomly select 3 rows
sampled_df = df.sample(3)
print(sampled_df)
```

Question 10: Handling MultiIndex (5 Marks)

```
import pandas as pd

# 1. Create DataFrame with MultiIndex
arrays = [
    ['North', 'North', 'South', 'South'],
    ['New York', 'Chicago', 'Miami', 'Houston']
]
index = pd.MultiIndex.from_arrays(arrays, names=('Region', 'City'))
data = {
    'Population': [8500000, 2700000, 470000, 2300000]
}
df = pd.DataFrame(data, index=index)

# 2. Reset the MultiIndex
df_reset = df.reset_index()
print(df_reset)
```