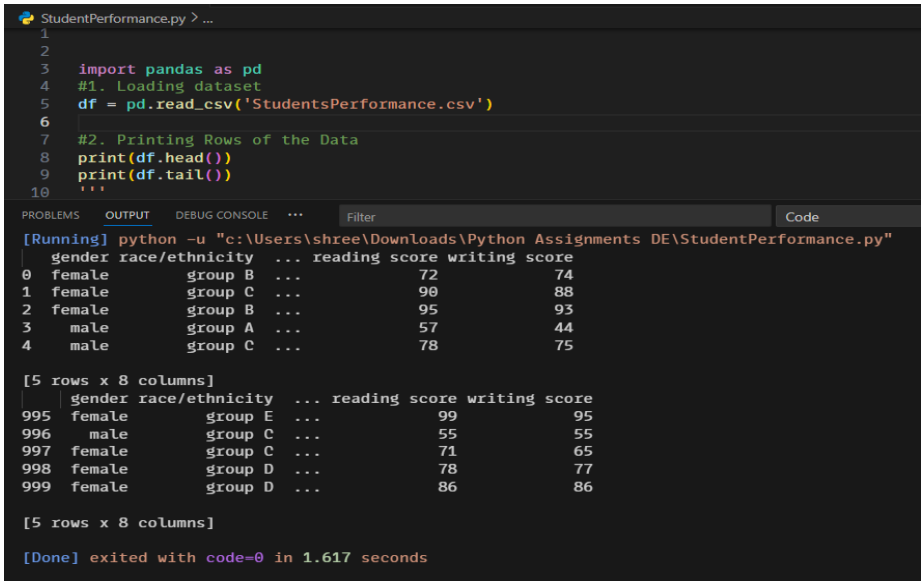
DE PYTHON CODING CHALLENGE

I) Execute Data cleaning programs:

1. Loading the dataset and printing rows for data



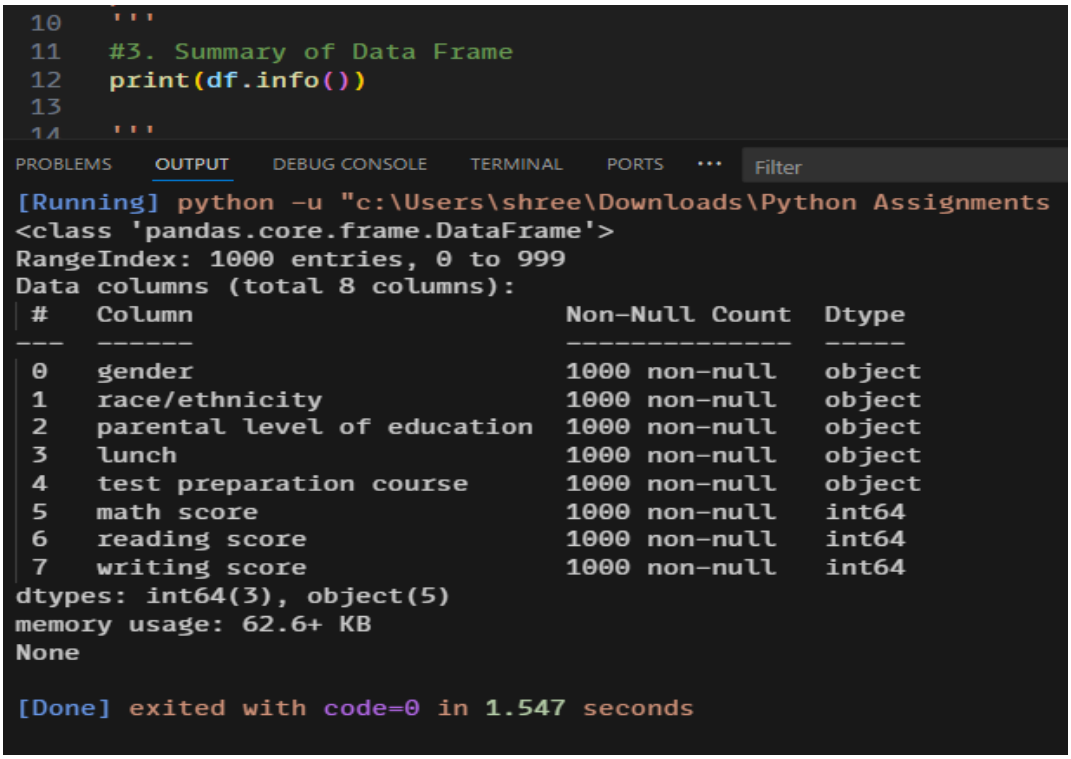
```
StudentPerformance.py > ...
1
2
3   import pandas as pd
4   #1. Loading dataset
5   df = pd.read_csv('StudentsPerformance.csv')
6
7   #2. Printing Rows of the Data
8   print(df.head())
9   print(df.tail())
10  '''
PROBLEMS  OUTPUT  DEBUG CONSOLE  ...  Filter  Code
[Running] python -u "c:\Users\shree\Downloads\Python Assignments DE\StudentPerformance.py"
  gender race/ethnicity  ... reading score writing score
0  female        group B  ...            72            74
1  female        group C  ...            90            88
2  female        group B  ...            95            93
3    male        group A  ...            57            44
4    male        group C  ...            78            75

[5 rows x 8 columns]
     gender race/ethnicity  ... reading score writing score
995  female        group E  ...            99            95
996    male        group C  ...            55            55
997  female        group C  ...            71            65
998  female        group D  ...            78            77
999  female        group D  ...            86            86

[5 rows x 8 columns]

[Done] exited with code=0 in 1.617 seconds
```

2. Summary of data frame using info( )



```
10  '''
11  #3. Summary of Data Frame
12  print(df.info())
13
14  '''
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  ...  Filter
[Running] python -u "c:\Users\shree\Downloads\Python Assignments
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 8 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   gender                       1000 non-null   object
 1   race/ethnicity               1000 non-null   object
 2   parental level of education  1000 non-null   object
 3   lunch                        1000 non-null   object
 4   test preparation course      1000 non-null   object
 5   math score                   1000 non-null   int64
 6   reading score                1000 non-null   int64
 7   writing score                1000 non-null   int64
dtypes: int64(3), object(5)
memory usage: 62.6+ KB
None

[Done] exited with code=0 in 1.547 seconds
```

3. **Descriptive statistical measures using describe( )**

```
'''
#4. Descriptive Statistical Measures
print(df.describe())

'''
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  ...  Filter

[Running] python -u "c:\Users\shree\Downloads\Python Assignments
       math score  reading score  writing score
count  1000.00000    1000.000000    1000.000000
mean     66.08900      69.169000      68.054000
std      15.16308      14.600192      15.195657
min       0.00000      17.000000      10.000000
25%      57.00000      59.000000      57.750000
50%      66.00000      70.000000      69.000000
75%      77.00000      79.000000      79.000000
max     100.00000     100.000000     100.000000

[Done] exited with code=0 in 1.428 seconds
```

4. **Counting null values in each column using isnull( )**

```
'''
#5. counting null values in each column
print(df.isnull().sum())

'''
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  ...  Filter

[Running] python -u "c:\Users\shree\Downloads\Python Assignments
gender                       1
race/ethnicity               0
parental level of education  2
lunch                        0
test preparation course      1
math score                   0
reading score                0
writing score                0
dtype: int64

[Done] exited with code=0 in 1.504 seconds
```

II) Panda joins in Python:

1. **Loading and cleaning the dataset, and also creating two subsets to perform joins**

```
#joins in pandas
import pandas as pd

# 1. Load and clean the dataset
df = pd.read_csv('StudentsPerformance.csv')
df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_')

# 2. Create two subsets of data with non-overlapping indexes

scores = df.loc[0:4, ['gender', 'math_score', 'reading_score', 'writing_score']].copy()
scores['student_id'] = scores.index

profile = df.loc[2:6, ['gender', 'race/ethnicity', 'test_preparation_course']].copy()
profile['student_id'] = profile.index

print("\n--- Scores DataFrame ---\n", scores)
print("\n--- Profile DataFrame ---\n", profile)
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  QUERY RESULTS  Filter

[Running] python -u "c:\Users\shree\Downloads\Python Assignments DE\StudentPerformance.py"

--- Scores DataFrame ---
   gender  math_score  reading_score  writing_score  student_id
0  female          72             72             74           0
1     NaN          69             90             88           1
2  female          90             95             93           2
3    male          47             57             44           3
4    male          76             78             75           4

--- Profile DataFrame ---
   gender race/ethnicity test_preparation_course  student_id
2  female        group B                    none           2
3    male        group A                    none           3
4    male        group C                     NaN           4
5  female        group B                    none           5
6  female        group B               completed           6
```

2. Performing Inner join and Left join

```
41  # 3. Perform all four types of joins
42
43  # INNER JOIN (only matching student_ids)
44  inner_join = pd.merge(scores, profile, on='student_id', how='inner', suffixes=('_score', '_profile'))
45  print("\n--- Inner Join ---\n", inner_join)
46
47  # LEFT JOIN (all from scores, match from profile if exists)
48  left_join = pd.merge(scores, profile, on='student_id', how='left', suffixes=('_score', '_profile'))
49  print("\n--- Left Join ---\n", left_join)
50
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS Filter

```
--- Inner Join ---
   gender_score  math_score  ...  race/ethnicity  test_preparation_course
0        female          90  ...         group B                     none
1          male          47  ...         group A                     none
2          male          76  ...         group C                      NaN

[3 rows x 8 columns]

--- Left Join ---
   gender_score  math_score  ...  race/ethnicity  test_preparation_course
0        female          72  ...             NaN                      NaN
1           NaN          69  ...             NaN                      NaN
2        female          90  ...         group B                     none
3          male          47  ...         group A                     none
4          male          76  ...         group C                      NaN

[5 rows x 8 columns]
```

3. Performing Right join and Outer join

```
50
51  # RIGHT JOIN (all from profile, match from scores if exists)
52  right_join = pd.merge(scores, profile, on='student_id', how='right', suffixes=('_score', '_profile'))
53  print("\n--- Right Join ---\n", right_join)
54
55  # OUTER JOIN (all records from both sides, NaNs where no match)
56  outer_join = pd.merge(scores, profile, on='student_id', how='outer', suffixes=('_score', '_profile'))
57  print("\n--- Outer Join ---\n", outer_join)
58
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS Filter

```
--- Right Join ---
   gender_score  math_score  ...  race/ethnicity  test_preparation_course
0        female        90.0  ...         group B                     none
1          male        47.0  ...         group A                     none
2          male        76.0  ...         group C                      NaN
3           NaN         NaN  ...         group B                     none
4           NaN         NaN  ...         group B                completed

[5 rows x 8 columns]

--- Outer Join ---
   gender_score  math_score  ...  race/ethnicity  test_preparation_course
0        female        72.0  ...             NaN                      NaN
1           NaN        69.0  ...             NaN                      NaN
2        female        90.0  ...         group B                     none
3          male        47.0  ...         group A                     none
4          male        76.0  ...         group C                      NaN
5           NaN         NaN  ...         group B                     none
6           NaN         NaN  ...         group B                completed

[7 rows x 8 columns]

[Done] exited with code=0 in 1.627 seconds
```