## Experiment No : 14

**Aim :** Write python programs to implement basic operations using pandas like series, data frames, indexing, filtering, combining and merging data frames.

**Description :**

PANDAS

1) high performance data analysis tool

2) working with large data set

3) supports load files with different formats

4) more flexible

5) represents in tabular way(rows and columns.

6)working on missing data

7) indexing-slicing-subsetting the large data sets

8)merge and join two different datasets easily

9)Reshape data sets

(3 types of data structures):

1) series(one dimensional) list

2)dataframes( two dimensional)list/dictionary/series/another dataframe

3)panel (multidimentional)data , col major axis, row minor axis

import pandas as pd

syntax

pd.Series(data,index)

pd.DataFrame(data) most efficient

pd.Panel(data, major axis, minor axis, dtype)

**Implementation :**

**Code :**

```
import pandas as pd
import numpy as np
l=[10,20,30,40]
print(pd.Series(l))
# change index value
print(pd.Series(l, index=['i','ii','iii','iv']))
d={'name':['shri', 'dhar', 'shridhar'], 'percentage':[90,85,95]}
print(pd.DataFrame(d))
"""
# attributes series
index--series.index ---return all index values
array--series.array-----return an array of values
values-series.values----return values of series
name--series.name----return name and series
shape--series.shape--return the shape
ndim--series.ndim---return the dimensions of series
size--series.size---return the size and series
nbytes--series.nbytes---returns the memory occupied by values
memory_usage()--series.memory_usage()--return memory occupied by both index
and values
empty--series.empty--returns --True ,if series is empty,    False , if series
not empty
"""
import pandas as pd
import numpy as np
s=pd.Series([10,20,30,40,50],name="Numbers")
print(pd.Series(s))
print(s.index)
s=pd.Series([10,20,30,40,50],index=['a','b','c','d','e'],name="Numbers")
print(pd.Series(s))
print(s.index)
print(s.array)
```

```
print(s.values)
print(s.dtype)
s=pd.Series([10.5,20.5,30.5,40.5,50.5],index=['a','b','c','d','e'],name="Float
")
print(pd.Series(s))
print(s.index)
print(s.array)
print(s.values)
print(s.dtype)
print(s.shape)
print(s.ndim)
print(s.nbytes)
print(s.memory_usage()) # include index values
print(s.memory_usage(index=False))  # Gives memory occupied by values
print(s.size)
print(s.name)
print(s.empty)
s1=pd.Series()
print(s1.size)
print(s1)
print(s1.empty)
"""
# indexing/slicing
head()---->first 5 rows
head(no.of rows)----> 3 rows
tail()-----> last 5 rows
tail(no. of rows)---->4 rows
describe()------> count      avg
                  min        25%
                  max        75%
                  std
shape----no.of rows & no. of column
[start :   stop :    step]
[ 0    :max. length    1]
[                    default]
data-frame['column_name']--single
data-frame[[col1,col2]]----multiple column
data-frame[[col1,col2]][start:stop:step]

shape----no. of rows and columns
for index,row in df.iterrows():
        print(index,row)
df.loc[1]---row ( show all data of row 1)
df.loc[df[col-name]=='val'] (show data particilar column )
"""
import pandas as pd
import numpy as np
#pip install xlrd
#pip install openpyxl (xlsx file)
"""
Pandas uses the xlrd as their default engine for reading excel files.
However, xlrd has removed support for anything other than xls files in
```

```
their latest release. ... Install openpyxl: This is another
 excel package that still supports the xlsx format. Set the engine
to “openpyxl” instead of the default “xlrd”


"""
d=pd.read_excel("C:\\Users\\LENOVO\\Desktop\\python exps\\Book1.xlsx")
df=pd.DataFrame(d)
print(df)
print(df.head())
print(df.head(7))
print(df.tail())
print(df.tail(10))
print(df.describe())
print(df.columns)
print(df.shape)
print(df[['roll no.', 'Hindi']].head())
#slicing
print(df[1:10:2])
print(df[['roll no.','Name of student','Hindi']][1:6])
print(df.loc[1])  # show data of row 1)
print(df.loc[1:6])  # show data of all rows from 1 to 6
for index,rows in df.iterrows():
     print(index,rows)
for index,rows in df.iterrows():
     print(index,rows[['roll no.','Name of student','Hindi']])

print(df.loc[df['roll no.']==105]) #print , row of roll no 105
print(df.iloc[0:4,1:5])  # print, 0 to 3 index and 1 to 5 row
print(df.loc[0:4])  # print, 0 to 4 index



# data filtering
import pandas as pd
d=pd.read_excel("C:\\Users\\LENOVO\\Desktop\\python exps\\Book1.xlsx")
df=pd.DataFrame(d)
print(df)
print(df.loc[df['Hindi']<93])
print(df.loc[df['English']<93])
print(df.loc[(df["Hindi"]<92) & (df["English"]<92)])
print(df.loc[(df["Hindi"]<92) & (df["English"]<92) &  (df["Math"]<92)])
print(df.loc[df["Name of student"].str.contains("i")])
print(df.loc[~df["Name of student"].str.contains("i")])
print(df.loc[df["Name of student"].str.startswith("s")])
print(df.loc[~df["Name of student"].str.startswith("s")])
print(df.loc[df["Name of student"].str.endswith("s")])
print(df.loc[~df["Name of student"].str.endswith("s")])
"""
# PD.MERGE()
FOR COMBINING DATA ON COMMON COLUMNS
MORE FLEXIBLE. BUT BALSO COMPLEX OF THE METHODS WE WILL DISCUSS
```

```
MANY TO ONE AND MANY TO MANY JOINS ARE POSSIBLE
SIDE BY SIDE MERGE
"""
import pandas as pd


df1 = pd.DataFrame({'ID':pd.Series([1,2,3,5,9]),
                    'COL1':pd.Series([1,2,3,4,5]),
                    'COL2':pd.Series([6,7,8,9,10]),
                     'COL3':pd.Series([11,12,13,14,15]),
                      'COL4':pd.Series(['APPLE', 'ORANGE','BANANA',
'STRAWBERRY','CRANBERRY'])
                   })
df2 = pd.DataFrame({'ID':pd.Series([1,1,3,5]),
                    'COLA':pd.Series([8,9,10,11]),
                    'COLB':pd.Series([12,13,15,17]),
                    'COL4':pd.Series(['APPLE', 'ORANGE',
'STRAWBERRY','CRANBERRY']),
                     'COL4':pd.Series([1,2,3,4])
                   })
df3=df1.append(df2)
print(df3)

#print(df1)
#print(df2)
df3=df1.merge(df2, on='ID')
print(df3)
df3=df1.merge(df2, on='ID',how='inner')
print(df3)
df3=df1.merge(df2, on='ID',how='left')
print(df3)
df3=df1.merge(df2, on='ID',how='right')
print(df3)
df3=df1.merge(df2, on='ID',how='outer')
print(df3)
df3=pd.concat([df1,df2])
print(df3)
```

# Output :

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\LENOVO> python -u "c:\Users\LENOVO\Desktop\python exps\exp14.py"
0    10
1    20
2    30
3    40
dtype: int64
i      10
ii     20
iii    30
iv     40
dtype: int64
       name  percentage
0      shri          90
1      dhar          85
2  shridhar          95
0    10
1    20
2    30
3    40
4    50
Name: Numbers, dtype: int64
RangeIndex(start=0, stop=5, step=1)
a    10
b    20
c    30
d    40
e    50
Name: Numbers, dtype: int64
Name: Numbers, dtype: int64
Index(['a', 'b', 'c', 'd', 'e'], dtype='object')
<PandasArray>
[10, 20, 30, 40, 50]
Length: 5, dtype: int64
[10 20 30 40 50]
int64
a    10.5
b    20.5
c    30.5
d    40.5
e    50.5
Name: Float, dtype: float64
Index(['a', 'b', 'c', 'd', 'e'], dtype='object')
<PandasArray>
[10.5, 20.5, 30.5, 40.5, 50.5]
Length: 5, dtype: float64
[10.5 20.5 30.5 40.5 50.5]
float64
(5,)
1
40
80
40
5
Float
False
c:\Users\LENOVO\Desktop\python exps\exp14.py:71: FutureWarning: The default dtype for empty Series will be 'object' instead of 'float64' in a future version. Specify a dtype explicit
ly to silence this warning.
  s1=pd.Series()
0
Series([], dtype: float64)
True
```

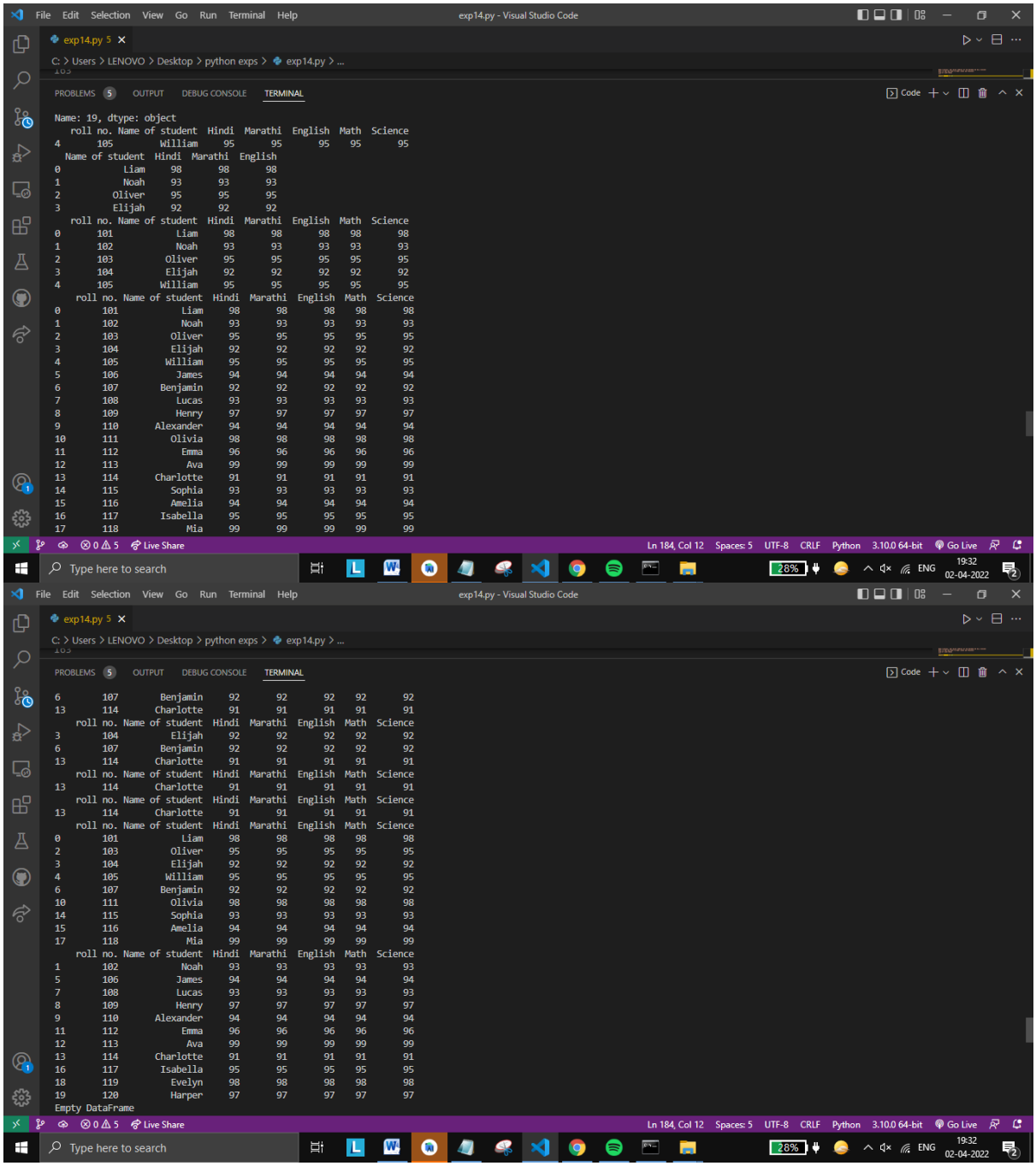
```
    roll no. Name of student  Hindi  Marathi  English  Math  Science
0        101           Liam     98       98       98    98       98
1        102           Noah     93       93       93    93       93
2        103         Oliver     95       95       95    95       95
3        104         Elijah     92       92       92    92       92
4        105        William     95       95       95    95       95
5        106          James     94       94       94    94       94
6        107       Benjamin     92       92       92    92       92
7        108          Lucas     93       93       93    93       93
8        109          Henry     97       97       97    97       97
9        110      Alexander     94       94       94    94       94
10       111         Olivia     98       98       98    98       98
11       112           Emma     96       96       96    96       96
12       113            Ava     99       99       99    99       99
13       114      Charlotte     91       91       91    91       91
14       115         Sophia     93       93       93    93       93
15       116         Amelia     94       94       94    94       94
16       117       Isabella     95       95       95    95       95
17       118            Mia     99       99       99    99       99
18       119         Evelyn     98       98       98    98       98
19       120         Harper     97       97       97    97       97
   roll no. Name of student  Hindi  Marathi  English  Math  Science
0      101           Liam     98       98       98    98       98
1      102           Noah     93       93       93    93       93
2      103         Oliver     95       95       95    95       95
3      104         Elijah     92       92       92    92       92
4      105        William     95       95       95    95       95
   roll no. Name of student  Hindi  Marathi  English  Math  Science
0      101           Liam     98       98       98    98       98
1      102           Noah     93       93       93    93       93
2      103         Oliver     95       95       95    95       95
3      104         Elijah     92       92       92    92       92
4      105        William     95       95       95    95       95
```

```
  4       105        William     95       95       95    95       95
  5       106          James     94       94       94    94       94
  6       107       Benjamin     92       92       92    92       92
     roll no. Name of student  Hindi  Marathi  English  Math  Science
  15      116         Amelia     94       94       94    94       94
  16      117       Isabella     95       95       95    95       95
  17      118            Mia     99       99       99    99       99
  18      119         Evelyn     98       98       98    98       98
  19      120         Harper     97       97       97    97       97
     roll no. Name of student  Hindi  Marathi  English  Math  Science
  10      111         Olivia     98       98       98    98       98
  11      112           Emma     96       96       96    96       96
  12      113            Ava     99       99       99    99       99
  13      114      Charlotte     91       91       91    91       91
  14      115         Sophia     93       93       93    93       93
  15      116         Amelia     94       94       94    94       94
  16      117       Isabella     95       95       95    95       95
  17      118            Mia     99       99       99    99       99
  18      119         Evelyn     98       98       98    98       98
  19      120         Harper     97       97       97    97       97
         roll no.      Hindi    Marathi    English       Math    Science
  count   20.00000  20.000000  20.000000  20.000000  20.000000  20.000000
  mean   110.50000  95.150000  95.150000  95.150000  95.150000  95.150000
  std      5.91608   2.476734   2.476734   2.476734   2.476734   2.476734
  min    101.00000  91.000000  91.000000  91.000000  91.000000  91.000000
  25%    105.75000  93.000000  93.000000  93.000000  93.000000  93.000000
  50%    110.50000  95.000000  95.000000  95.000000  95.000000  95.000000
  75%    115.25000  97.250000  97.250000  97.250000  97.250000  97.250000
  max    120.00000  99.000000  99.000000  99.000000  99.000000  99.000000
  Index(['roll no.', 'Name of student', 'Hindi', 'Marathi', 'English', 'Math',
         'Science'],
```

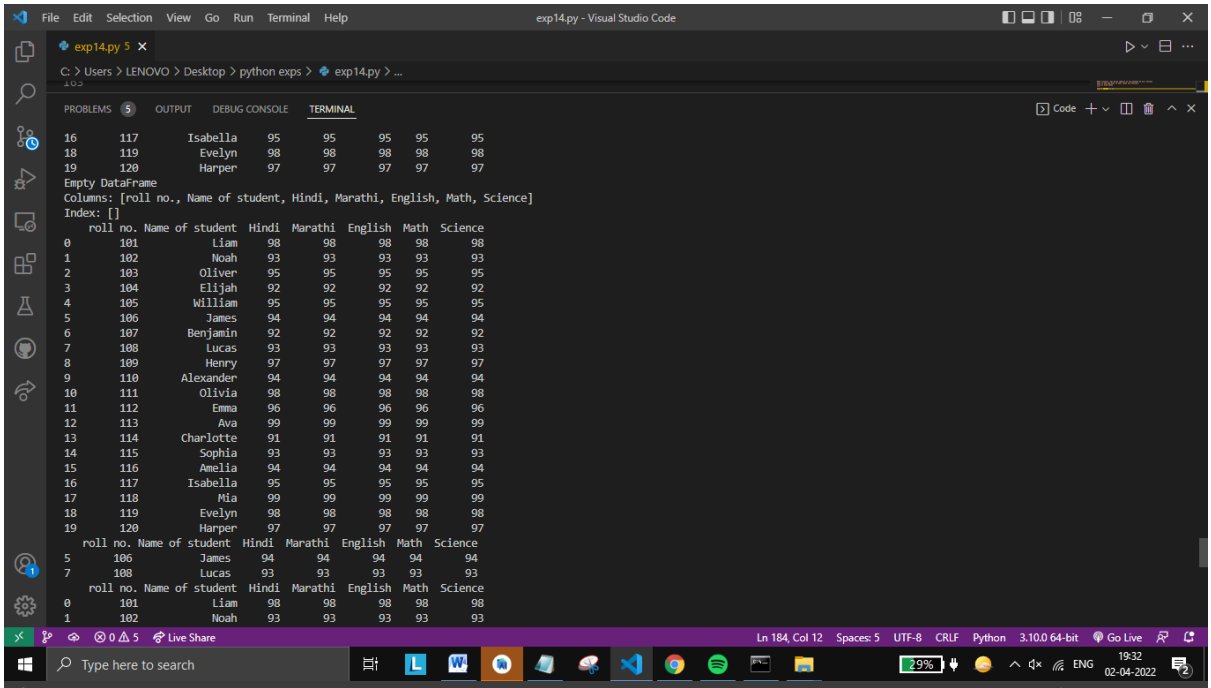
```
(20, 7)
   roll no.  Hindi
0       101     98
1       102     93
2       103     95
3       104     92
4       105     95
   roll no. Name of student  Hindi  Marathi  English  Math  Science
1       102            Noah     93       93       93    93       93
3       104          Elijah     92       92       92    92       92
5       106           James     94       94       94    94       94
7       108           Lucas     93       93       93    93       93
9       110       Alexander     94       94       94    94       94
   roll no. Name of student  Hindi
1       102            Noah     93
2       103          Oliver     95
3       104          Elijah     92
4       105         William     95
5       106           James     94
roll no.            102
Name of student    Noah
Hindi                93
Marathi              93
English              93
Math                 93
Science              93
Name: 1, dtype: object
   roll no. Name of student  Hindi  Marathi  English  Math  Science
1       102            Noah     93       93       93    93       93
2       103          Oliver     95       95       95    95       95
3       104          Elijah     92       92       92    92       92
4       105         William     95       95       95    95       95
5       106           James     94       94       94    94       94
```

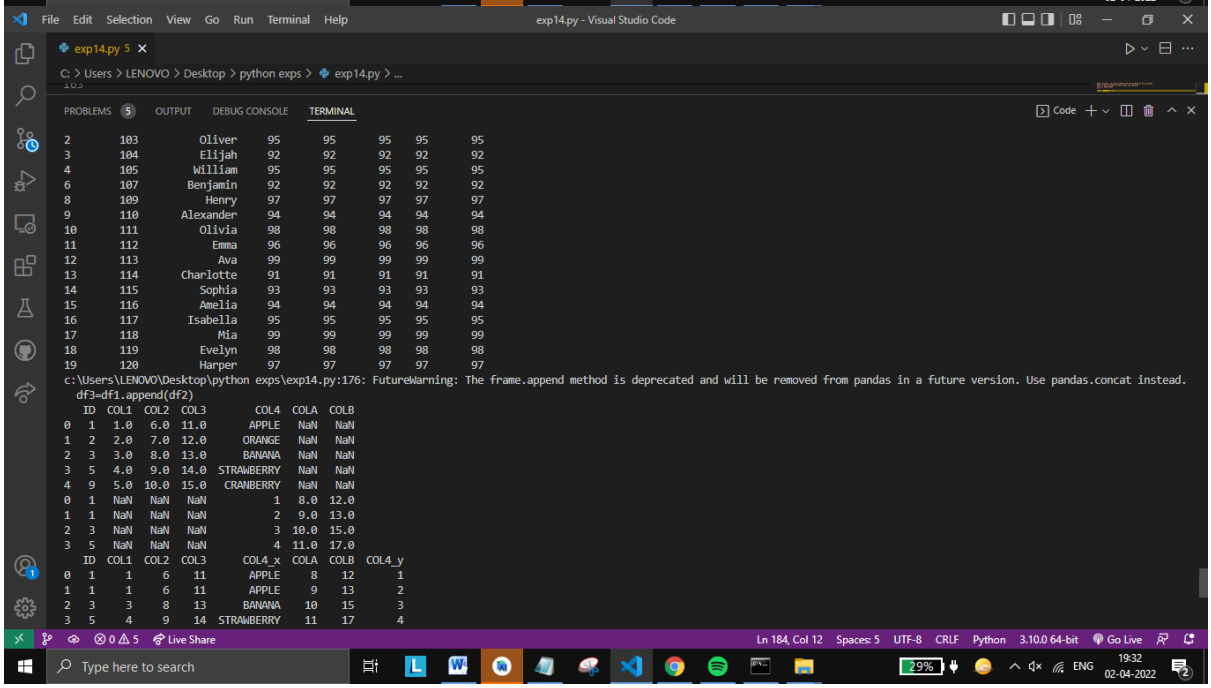
```
4       105        William     95       95       95    95       95
5       106          James     94       94       94    94       94
6       107        Benjamin    92       92       92    92       92
0 roll no.            101
Name of student     Liam
Hindi                 98
Marathi               98
English               98
Math                  98
Science               98
Name: 0, dtype: object
1 roll no.            102
Name of student     Noah
Hindi                 93
Marathi               93
English               93
Math                  93
Science               93
Name: 1, dtype: object
2 roll no.             103
Name of student     Oliver
Hindi                 95
Marathi               95
English               95
Math                  95
Science               95
Name: 2, dtype: object
3 roll no.             104
Name of student     Elijah
Hindi                 92
Marathi               92
English               92
Math                  92
```

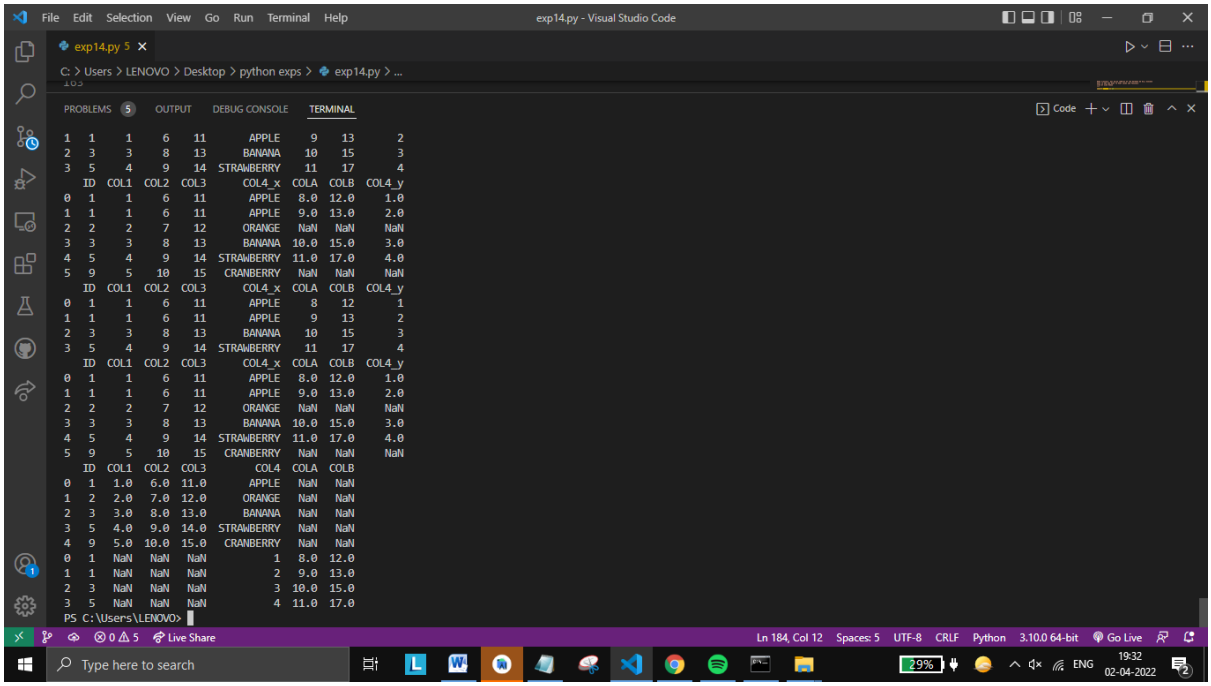
```
Name: 4, dtype: object
5 roll no.             106
Name of student    James
Hindi                 94
Marathi               94
English               94
Math                  94
Science               94
Name: 5, dtype: object
6 roll no.              107
Name of student    Benjamin
Hindi                    92
Marathi                  92
English                  92
Math                     92
Science                  92
Name: 6, dtype: object
7 roll no.             108
Name of student    Lucas
Hindi                 93
Marathi               93
English               93
Math                  93
Science               93
Name: 7, dtype: object
8 roll no.             109
Name of student    Henry
Hindi                 97
Marathi               97
English               97
Math                  97
Science               97
Name: 8, dtype: object
```


File Edit Selection View Go Run Terminal Help
exp14.py - Visual Studio Code
exp14.py 5
C: > Users > LENOVO > Desktop > python exps > exp14.py > ...
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL
Code
Name: 19, dtype: object
roll no. Name of student Hindi Marathi English Math Science
4 105 William 95 95 95 95 95
Name of student Hindi Marathi English
0 Liam 98 98 98
1 Noah 93 93 93
2 Oliver 95 95 95
3 Elijah 92 92 92
roll no. Name of student Hindi Marathi English Math Science
0 101 Liam 98 98 98 98 98
1 102 Noah 93 93 93 93 93
2 103 Oliver 95 95 95 95 95
3 104 Elijah 92 92 92 92 92
4 105 William 95 95 95 95 95
roll no. Name of student Hindi Marathi English Math Science
0 101 Liam 98 98 98 98 98
1 102 Noah 93 93 93 93 93
2 103 Oliver 95 95 95 95 95
3 104 Elijah 92 92 92 92 92
4 105 William 95 95 95 95 95
5 106 James 94 94 94 94 94
6 107 Benjamin 92 92 92 92 92
7 108 Lucas 93 93 93 93 93
8 109 Henry 97 97 97 97 97
9 110 Alexander 94 94 94 94 94
10 111 Olivia 98 98 98 98 98
11 112 Emma 96 96 96 96 96
12 113 Ava 99 99 99 99 99
13 114 Charlotte 91 91 91 91 91
14 115 Sophia 93 93 93 93 93
15 116 Amelia 94 94 94 94 94
16 117 Isabella 95 95 95 95 95
17 118 Mia 99 99 99 99 99
Ln 184, Col 12 Spaces: 5 UTF-8 CRLF Python 3.10.0 64-bit Go Live
Live Share
Type here to search
28%
ENG
19:32
02-04-2022


File Edit Selection View Go Run Terminal Help
exp14.py - Visual Studio Code
exp14.py 5
C: > Users > LENOVO > Desktop > python exps > exp14.py > ...
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL
Code
6 107 Benjamin 92 92 92 92 92
13 114 Charlotte 91 91 91 91 91
roll no. Name of student Hindi Marathi English Math Science
3 104 Elijah 92 92 92 92 92
6 107 Benjamin 92 92 92 92 92
13 114 Charlotte 91 91 91 91 91
roll no. Name of student Hindi Marathi English Math Science
13 114 Charlotte 91 91 91 91 91
roll no. Name of student Hindi Marathi English Math Science
13 114 Charlotte 91 91 91 91 91
roll no. Name of student Hindi Marathi English Math Science
0 101 Liam 98 98 98 98 98
2 103 Oliver 95 95 95 95 95
3 104 Elijah 92 92 92 92 92
4 105 William 95 95 95 95 95
6 107 Benjamin 92 92 92 92 92
10 111 Olivia 98 98 98 98 98
14 115 Sophia 93 93 93 93 93
15 116 Amelia 94 94 94 94 94
17 118 Mia 99 99 99 99 99
roll no. Name of student Hindi Marathi English Math Science
1 102 Noah 93 93 93 93 93
5 106 James 94 94 94 94 94
7 108 Lucas 93 93 93 93 93
8 109 Henry 97 97 97 97 97
9 110 Alexander 94 94 94 94 94
11 112 Emma 96 96 96 96 96
12 113 Ava 99 99 99 99 99
13 114 Charlotte 91 91 91 91 91
16 117 Isabella 95 95 95 95 95
18 119 Evelyn 98 98 98 98 98
19 120 Harper 97 97 97 97 97
Empty DataFrame
Ln 184, Col 12 Spaces: 5 UTF-8 CRLF Python 3.10.0 64-bit Go Live
Live Share
Type here to search
28%
ENG
19:32
02-04-2022


File Edit Selection View Go Run Terminal Help
exp14.py - Visual Studio Code
exp14.py 5
C: > Users > LENOVO > Desktop > python exps > exp14.py > ...
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL
16 117 Isabella 95 95 95 95 95
18 119 Evelyn 98 98 98 98 98
19 120 Harper 97 97 97 97 97
Empty DataFrame
Columns: [roll no., Name of student, Hindi, Marathi, English, Math, Science]
Index: []
roll no. Name of student Hindi Marathi English Math Science
0 101 Liam 98 98 98 98 98
1 102 Noah 93 93 93 93 93
2 103 Oliver 95 95 95 95 95
3 104 Elijah 92 92 92 92 92
4 105 William 95 95 95 95 95
5 106 James 94 94 94 94 94
6 107 Benjamin 92 92 92 92 92
7 108 Lucas 93 93 93 93 93
8 109 Henry 97 97 97 97 97
9 110 Alexander 94 94 94 94 94
10 111 Olivia 98 98 98 98 98
11 112 Emma 96 96 96 96 96
12 113 Ava 99 99 99 99 99
13 114 Charlotte 91 91 91 91 91
14 115 Sophia 93 93 93 93 93
15 116 Amelia 94 94 94 94 94
16 117 Isabella 95 95 95 95 95
17 118 Mia 99 99 99 99 99
18 119 Evelyn 98 98 98 98 98
19 120 Harper 97 97 97 97 97
roll no. Name of student Hindi Marathi English Math Science
5 106 James 94 94 94 94 94
7 108 Lucas 93 93 93 93 93
roll no. Name of student Hindi Marathi English Math Science
0 101 Liam 98 98 98 98 98
1 102 Noah 93 93 93 93 93
Ln 184, Col 12 Spaces: 5 UTF-8 CRLF Python 3.10.0 64-bit Go Live
Type here to search
29%
ENG
19:32
02-04-2022


File Edit Selection View Go Run Terminal Help
exp14.py - Visual Studio Code
exp14.py 5
C: > Users > LENOVO > Desktop > python exps > exp14.py > ...
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL
2 103 Oliver 95 95 95 95 95
3 104 Elijah 92 92 92 92 92
4 105 William 95 95 95 95 95
6 107 Benjamin 92 92 92 92 92
8 109 Henry 97 97 97 97 97
9 110 Alexander 94 94 94 94 94
10 111 Olivia 98 98 98 98 98
11 112 Emma 96 96 96 96 96
12 113 Ava 99 99 99 99 99
13 114 Charlotte 91 91 91 91 91
14 115 Sophia 93 93 93 93 93
15 116 Amelia 94 94 94 94 94
16 117 Isabella 95 95 95 95 95
17 118 Mia 99 99 99 99 99
18 119 Evelyn 98 98 98 98 98
19 120 Harper 97 97 97 97 97
c:\Users\LENOVO\Desktop\python exps\exp14.py:176: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
df3=df1.append(df2)
ID COL1 COL2 COL3 COL4 COLA COLB
0 1 1.0 6.0 11.0 APPLE NaN NaN
1 2 2.0 7.0 12.0 ORANGE NaN NaN
2 3 3.0 8.0 13.0 BANANA NaN NaN
3 5 4.0 9.0 14.0 STRAWBERRY NaN NaN
4 9 5.0 10.0 15.0 CRANBERRY NaN NaN
0 1 NaN NaN NaN 1 8.0 12.0
1 1 NaN NaN NaN 2 9.0 13.0
2 3 NaN NaN NaN 3 10.0 15.0
3 5 NaN NaN NaN 4 11.0 17.0
ID COL1 COL2 COL3 COL4_x COLA COLB COL4_y
0 1 1 6 11 APPLE 8 12 1
1 1 1 6 11 APPLE 9 13 2
2 3 3 8 13 BANANA 10 15 3
3 5 4 9 14 STRAWBERRY 11 17 4
Ln 184, Col 12 Spaces: 5 UTF-8 CRLF Python 3.10.0 64-bit Go Live
Type here to search
29%
ENG
19:32
02-04-2022




**Conclusion :** Therefore we have successfully implemented basic operations using pandas like series, data frames, indexing, filtering, combining and merging data frames.