Assessment

I am going to provide two .csv files , you are supposed to work on them and have to provide solutions to the following problems

import necessary libraries

In [11]:
```
import pandas as pd
```

merge those two csv files (after getting as dataframes, get them as a single dataframe)

In [20]:
```
df = pd.concat(
    map(pd.read_csv, ['college_1.csv', 'college_2.csv']), ignore_index=True)
print(df)
```

```
              Name  python  mysql  Previous Geekions  CodeKata Score  \
0        A.Dharani    82.0   20.0              24500           24500
1      V.JEEVITHA     82.0   20.0              21740           21740
2      HEMAVATHI.R   100.0  100.0              19680           19680
3      Mugunthan S   100.0   47.0              10610           10610
4      Sathammai.S   100.0    8.0               8980            8980
..             ...     ...    ...                ...             ...
114  praveen raj j    24.0    0.0               2380            2380
115     AMARNATH D    -1.0   12.0               1890            1890
116           bala    32.0    0.0               1720            1720
117           XY Z    -1.0   -1.0                  0               0
118      Hariharan    -1.0   -1.0                  0               0

                                  Department  Rising  python_en  \
0            Computer Science and Engineering       0        NaN
1            Computer Science and Engineering       0        NaN
2            Computer Science and Engineering       0        NaN
3            Computer Science and Engineering       0        NaN
4            Computer Science and Engineering       0        NaN
..                                        ...     ...        ...
114          Computer Science and Engineering       0       -1.0
115  Electronics and Communication Engineering       0       52.0
116  Electronics and Communication Engineering       0       49.0
117          Computer Science and Engineering       0       20.0
118          Computer Science and Engineering       0       -1.0

     computational_thinking
0                       NaN
1                       NaN
2                       NaN
3                       NaN
4                       NaN
..                      ...
114                     0.0
115                    -1.0
116                    -1.0
117                    -1.0
118                     0.0

[119 rows x 9 columns]
```

Take each csv file , split that csv file into multiple categories (example csv files are added in the repo)

consider if the codekata score exceeds 15000 points(present week) then make a csv on those observations as Exceeded expectations.csv

if 10000<codekata score<15000 (Reached_expectations.csv)

if 7000<codekata score<10000 (Needs_Improvement.csv)

if codekate score < 7000 (Unsatisfactory.csv)

In [13]:
```
['Unsatisfactory.csv' if i < 7000 else 'Needs_Improvement.csv' if i > 10000 else 'Reached_expecta
```

```
Out[13]: ['Needs_Improvement.csv',
 'Needs_Improvement.csv',
 'Needs_Improvement.csv',
 'Needs_Improvement.csv',
 'Reached_expectations.csv',
 'Reached_expectations.csv',
 'Reached_expectations.csv',
 'Reached_expectations.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
```

```
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Needs_Improvement.csv',
 'Needs_Improvement.csv',
 'Needs_Improvement.csv',
 'Needs_Improvement.csv',
 'Reached_expectations.csv',
 'Reached_expectations.csv',
 'Reached_expectations.csv',
 'Reached_expectations.csv',
 'Reached_expectations.csv',
 'Reached_expectations.csv',
 'Reached_expectations.csv',
 'Reached_expectations.csv',
 'Reached_expectations.csv',
 'Reached_expectations.csv',
 'Reached_expectations.csv',
 'Reached_expectations.csv',
 'Reached_expectations.csv',
 'Reached_expectations.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv',
 'Unsatisfactory.csv']
```

Average of previous week geekions vs this week geekions (i.e Previous Geekions vs CodeKata Score)

In [14]:
```
df.groupby(['python','mysql','python_en','computational_thinking']).mean()
```

```
C:\Users\Karthi\AppData\Local\Temp\ipykernel_10424\4187661029.py:1: FutureWarning: The default v
alue of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version, numeric_only w
ill default to False. Either specify numeric_only or select only columns which should be valid f
or the function.
  df.groupby(['python','mysql','python_en','computational_thinking']).mean()
```

Out[14]:

python	mysql	python_en	computational_thinking	Previous Geekions	CodeKata Score	Rising
-1.0	-1.0	-1.0	0.0	0.000000	0.000000	0.000000
		20.0	-1.0	0.000000	0.000000	0.000000
	0.0	-1.0	0.0	4800.000000	6800.000000	2000.000000
		0.0	0.0	5616.666667	6583.333333	966.666667
			3.0	7250.000000	8950.000000	1700.000000
		9.0	0.0	7670.000000	8050.000000	380.000000
		20.0	0.0	5290.000000	6290.000000	1000.000000
		40.0	-1.0	5050.000000	5050.000000	0.000000
		43.0	0.0	3980.000000	5280.000000	1300.000000
		46.0	0.0	5200.000000	5200.000000	0.000000
		52.0	0.0	3860.000000	4440.000000	580.000000
	4.0	0.0	6.0	4020.000000	4020.000000	0.000000
		6.0	0.0	5300.000000	6640.000000	1340.000000
		60.0	6.0	8650.000000	8650.000000	0.000000
	12.0	52.0	-1.0	1890.000000	1890.000000	0.000000
	20.0	100.0	0.0	6170.000000	8160.000000	1990.000000
	24.0	55.0	6.0	8790.000000	10790.000000	2000.000000
		100.0	-1.0	9150.000000	9150.000000	0.000000
	31.0	63.0	0.0	6710.000000	7550.000000	840.000000
	35.0	0.0	-1.0	10040.000000	10040.000000	0.000000
		55.0	6.0	3220.000000	3220.000000	0.000000
		72.0	39.0	7310.000000	7630.000000	320.000000
	62.0	15.0	9.0	7470.000000	8090.000000	620.000000
	100.0	23.0	-1.0	7170.000000	7730.000000	560.000000
0.0	0.0	20.0	0.0	14150.000000	14490.000000	340.000000
	35.0	78.0	0.0	7210.000000	8970.000000	1760.000000
16.0	24.0	20.0	0.0	6060.000000	6090.000000	30.000000
24.0	0.0	-1.0	0.0	2380.000000	2380.000000	0.000000
32.0	0.0	49.0	-1.0	1720.000000	1720.000000	0.000000
58.0	0.0	0.0	0.0	5180.000000	8320.000000	3140.000000
100.0	0.0	0.0	0.0	7340.000000	8030.000000	690.000000
	31.0	0.0	9.0	19400.000000	19400.000000	0.000000

No of students participated

In [15]:
```
#Grouping and perform count over each group
df =  df.groupby('Department')['Department'].count()
```

```
print(df)
```

```
Department
Computer Science and Engineering             63
Electronics and Communication Engineering    39
Electronics and Electrical Engineering       17
Name: Department, dtype: int64
```

#Average completion of python course or my_sql or python english or computational thinking

In [21]:
```
df.groupby(['Previous Geekions','CodeKata Score']).mean()
```

```
C:\Users\Karthi\AppData\Local\Temp\ipykernel_10424\556006490.py:1: FutureWarning: The default va
lue of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version, numeric_only wi
ll default to False. Either specify numeric_only or select only columns which should be valid fo
r the function.
  df.groupby(['Previous Geekions','CodeKata Score']).mean()
```

Out[21]:

		python	mysql	Rising	python_en	computational_thinking
Previous Geekions	**CodeKata Score**					
0	**0**	22.000000	3.571429	0.0	9.5	-0.5
40	**40**	98.333333	36.000000	0.0	NaN	NaN
60	**60**	82.000000	12.000000	0.0	NaN	NaN
100	**100**	65.000000	10.000000	0.0	NaN	NaN
120	**120**	94.875000	6.000000	0.0	NaN	NaN
...	**...**	...	...	...	...	...
14150	**14490**	0.000000	0.000000	340.0	20.0	0.0
19400	**19400**	100.000000	31.000000	0.0	0.0	9.0
19680	**19680**	100.000000	100.000000	0.0	NaN	NaN
21740	**21740**	82.000000	20.000000	0.0	NaN	NaN
24500	**24500**	82.000000	20.000000	0.0	NaN	NaN

103 rows × 5 columns

rising star of the week (top 3 candidate who performed well in that particular week)

In [22]:
```
df.sort_values('Rising',ascending=False,inplace=True)
(df.head(3))
```

Out[22]:

	Name	python	mysql	Previous Geekions	CodeKata Score	Department	Rising	python_en	computational_thinking
92	shifak N	58.0	0.0	5180	8320	Electronics and Electrical Engineering	3140	0.0	0.0
102	Narasimhan Y L	-1.0	0.0	4800	6800	Computer Science and Engineering	2000	-1.0	0.0
86	Ganesh Ramkumar R	-1.0	24.0	8790	10790	Computer Science and Engineering	2000	55.0	6.0

Shining stars of the week (top 3 candidates who has highest geekions)

In [23]:
```
df.sort_values('Previous Geekions',ascending=False,inplace=True)
(df.head(3))
```

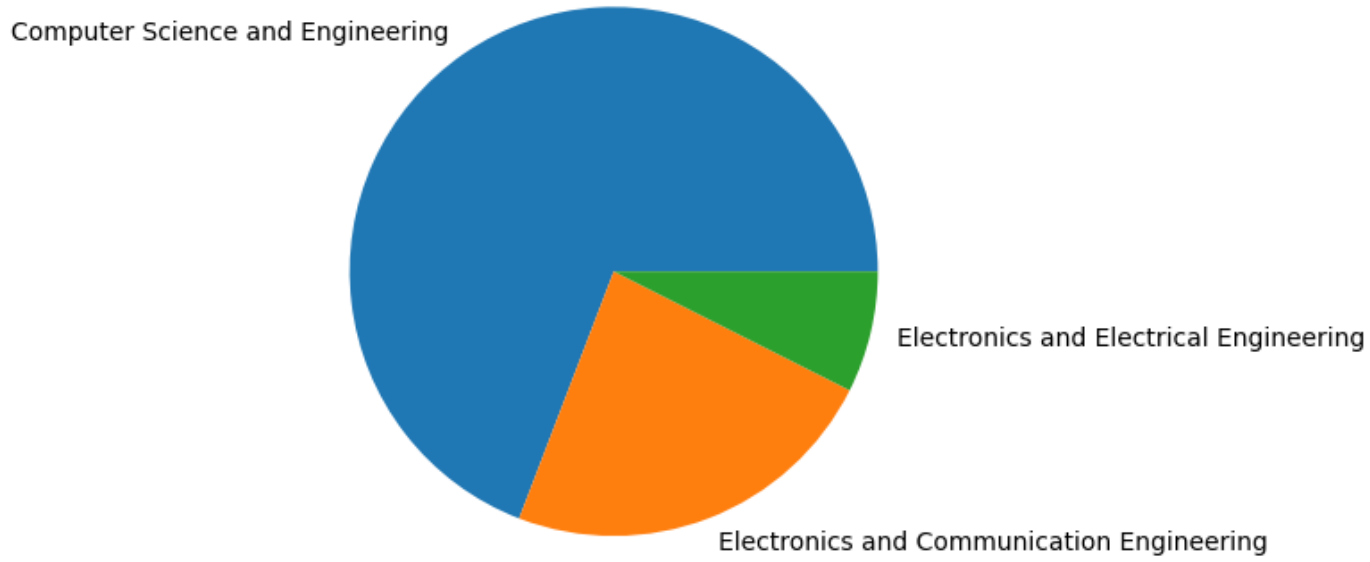
Out[23]:

	Name	python	mysql	Previous Geekions	CodeKata Score	Department	Rising	python_en	computational_thinking
0	A.Dharani	82.0	20.0	24500	24500	Computer Science and Engineering	0	NaN	NaN
1	V.JEEVITHA	82.0	20.0	21740	21740	Computer Science and Engineering	0	NaN	NaN
2	HEMAVATHI.R	100.0	100.0	19680	19680	Computer Science and Engineering	0	NaN	NaN

Department wise codekata performence (pie chart)

In [18]:
```
import matplotlib.pyplot as plt
dataFrame = pd.DataFrame({
    'Department' : ['Computer Science and Engineering' , 'Electronics and Communication Engineer
    'CodeKata_Score' : [320025,108335,34320]
})

plt.pie(dataFrame["CodeKata_Score"], labels = dataFrame["Department"])
plt.show()
```



Department wise toppers (horizantal bar graph or any visual representations of your choice)

In [25]:
```
import matplotlib.pyplot as plt

pd.pivot_table(index='Department', columns='CodeKata Score').plot(kind='bar', figsize=(15, 8))

plt.xlabel('Department')
plt.ylabel('CodeKata Score')
plt.title('Department wise toppers')
```

```
plt.legend(loc='upper right')
plt.show()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In [25], line 3
      1 import matplotlib.pyplot as plt
----> 3 pd.pivot_table(index='Department', columns='CodeKata Score').plot(kind='bar', figsize=(1
5, 8))
      5 plt.xlabel('Department')
      6 plt.ylabel('CodeKata Score')

TypeError: pivot_table() missing 1 required positional argument: 'data'
```

In [ ]:

In [ ]: