First Lecture on Pandas

Create Read Update Delete (CRUD)

```
import pandas as pd
import numpy as np

#CREATE
data = {'Name': ['Aishwarya', 'Bhushan',
'Chetan','Dhananjay','Eknath','Faiz','Ganesh','NA'],
        'Age': [25,30,35,40,45,50,55,None], #np.nan
        'City':
['Surat','Mumbai','Pune','Nagpur','Chennai','Delhi','Kolka
ta','NA'],
        'Status':['Y','N','Y','N','Y','Y','N','NA']}

df = pd.DataFrame(data)

#READ
df.to_csv(r"C:\Users\YourName\Documents\Parag.csv",
index=False)   # Windows
#df.to_csv(r"/Users/YourName/Documents/Parag.csv",
index=False)     # Mac/Linux

df.to_csv('Parag.csv', index=False)
df = pd.read_excel("data.xlsx")
k=pd.read_csv('Parag.csv')
print(k)
#print(df)

#print(df.head(2))
#print(df.tail(3))
#print(df.describe(include='all'))  # Summary statistics
#print(df['Age'].mean())
#print(df['Status'].value_counts())

print(df.dropna(inplace=False))
#print(df)
#df['Values']=df['Age']+10
#print(df)
#print(k)
```

2nd Lecture on Pandas

```
import pandas as pd
import numpy as np

data = {'Name': ['Aishwarya', 'Bhushan',
'Chetan','Dhananjay','Eknath','Faiz','Ganesh','NA'],
        'Age': [25,30,35,40,45,50,55,None], #np.nan
        'City': ['Surat','Mumbai','Pune','Surat','Chennai','Delhi','Kolkata','NA'],
        'Status':['Y','N','Y','N','Y','Y','N','NA']}

df = pd.DataFrame(data)

#print(df.columns)
#print(df.index)
print(df.iloc[1,:])
df.loc[0,"Name"]=3
df.loc[df["Name"]=="Chetan","Age"]=30
df.loc[df["Name"]=="Chetan","Name"]=30
print(df)
#print(df.describe(include='all'))  # Summary statistics df.info()
print(df['Age'].mean())
print(df['Status'].value_counts())
#print(df.dropna(inplace=False))
df.drop("City", axis=1, inplace=True)  #2nd lecture
df.drop(0, axis=0, inplace=True)       #2nd lecture
df=df.drop(3)        #Similarly, fillna command can be used
print(df)
#df['Values']=df['Age']+10
df['Age']=df['Age']+10              #2nd lecture
#print(df)

print(df[df["City"]=="Surat"])
print(df[(df["City"]=="Surat") & (df["Status"]=="N")])
print(df[(df["City"]=="Surat") | (df["Status"]=="N")])
print(df[df["City"].str.startswith("S")])
print(df[df["City"].str.endswith("i")])
print(df[df["City"].isin(["Surat","Mumbai", "Delhi"])])
```

- **Important Commands**

Category	Common Commands	Purpose / Examples
1. Creation / Input-Output (I/O)	`DataFrame()`, `read_csv()`, `read_excel()`, `to_csv()`, `to_excel()`	Creating or loading data
2. Inspection / Overview	`head()`, `tail()`, `info()`, `describe()`, `shape`, `columns`, `index`	Basic summary and structure
3. Selection & Indexing	`loc[]`, `iloc[]`, `at[]`, `iat[]`, `[]`	Accessing rows, columns, and cells
4. Filtering & Conditional Selection	Boolean indexing (`df[df['Age'] > 30]`), `isin()`, `query()`	Selecting rows by condition
5. Modification / Updating	`assign()`, `replace()`, `rename()`, `astype()`, `map()`, `apply()`	Changing or updating data
6. Missing Data Handling	`isna()`, `notna()`, `fillna()`, `dropna()`, `interpolate()`	Handling NaN / missing values
7. Data Cleaning	`drop_duplicates()`, `sort_values()`, `reset_index()`, `set_index()`	Tidying and sorting data
8. Aggregation / Grouping	`groupby()`, `agg()`, `mean()`, `sum()`, `pivot_table()`	Summarizing data
9. Merging / Combining	`concat()`, `merge()`, `join()`, `append()`	Combining multiple DataFrames
10. Visualization	`plot()`, `hist()`, `boxplot()`	Quick visual analysis (built on Matplotlib)

3rd Lecture on Pandas

```
import pandas as pd
import numpy as np

data = {'Name': ['Aishwarya', 'Bhushan', 'Chetan', 'Dhananjay',
'Eknath', 'Faiz', 'Ganesh', None],
        'Age': [25, 30, np.nan, 40, 45, 50, None, 28],
        'City': ['Surat', 'Mumbai', 'Pune', 'Nagpur', 'Chennai',
None, 'Kolkata', 'Delhi'],
        'Status': ['Y', 'N', 'Y', 'N', 'Y', 'Y', 'N', 'Y']}

df = pd.DataFrame(data)

print("Original DataFrame:\n", df)

# Detect missing values
print("\nMissing values:\n", df.isnull().sum())

# Fill missing values
df['Age'] = df['Age'].fillna(df['Age'].mean())
df['City'] = df['City'].fillna('Unknown')
df['Name'] = df['Name'].fillna('Anonymous')
#print(df)

# Replace specific values
```

```
df['Status']=df['Status'].replace({'Y': 'Yes', 'N': 'No'})
#print(df)


# Rename columns
df=df.rename(columns={'Name': 'FullName', 'City': 'Location'})
#print(df)

# Change datatype
df['Age'] = df['Age'].astype(int)
print(df)

# Remove duplicates
df=df.drop_duplicates()
print(df)

# Sort values
#df=df.sort_values(by='Age', ascending=False)

#print("\nCleaned DataFrame:\n", df)
```

4th Lecture on Pandas

```
#7th CPC
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv('/Users/paragthakur/Desktop/Teaching Material/Data
Science for chemical Engineers/Salary.csv', header=2)
print(df.columns); print(df.head()); print(df.info());
print(df.describe())
print(df['1']); print(df.iloc[:,1]);
print(df[df['6']==47600])
print(df[df['6']!=47600])
print(df[(df['1']==35400) | (df['6']==35400)])
print(df[(df['1']<=25000) & (df['6']<=50000)])
print(df[~((df['1']<=25000) & (df['6']<=50000))])
print(df)
df.rename(columns={'1': 'Level 1','2':'Level 2','3':'Level 3',
                   '4':'Level 4','5':'Level 5','6':'Level 6',
                   '7':'Level 7','8':'Level 8','9':'Level 9',
                   '10':'Level 10','11':'Level 11','12':'Level 12',
                   '13':'Level 13','14':'Level 14','15':'Level 15',
                   '16':'Level 16','17':'Level 17','18':'Level 18',},
inplace=True)
#print(df.columns)
#print(df)
j=df.T
#print(j)
#print(df.max(axis=1))
(df.sort_values(by='3', ascending=False))
j=df.isna().max()
```

```
df.fillna(j, inplace=True)
#df[['1','3']].plot(kind='line',marker='*')
plt.scatter(df['Level'], df['1'], label='Stage 1', color='blue')
plt.scatter(df['Level'], df['3'], label='Stage 3', color='red')
plt.title("Comparison of Stage 1 and Stage 3 Pay Levels")
plt.xlabel("Stages of yearly Increment")
plt.ylabel("Basic Pay (₹)")
plt.show()
#print(df)
plt.show()
```

5th Lecture on Pandas

Focus Area	Key Commands
Grouping & Aggregation	`groupby()`, `agg()`, `mean()`, `sum()`
Combining Data	`merge()`, `join()`, `concat()`
Pivot & Advanced Analysis	`pivot_table()`, `corr()`, `cov()`

```
import pandas as pd

data = {'Department': ['HR', 'IT', 'IT', 'HR', 'Sales', 'Finance',
'Sales'],
        'Employee': ['A', 'B', 'C', 'D', 'E', 'F', 'G'],
        'Salary': [50000, 60000, 70000, 75000, 55000, 52000, 65000],
        'Experience': [3, 5, 7, 8, 2, 4, 6]}

df = pd.DataFrame(data)
#print("Original Data:\n", df)

# Group by department and get average salary
grouped = df.groupby('Department')
#print(grouped)
#print(grouped['Salary'].mean())  # Mean salary by department
#print(grouped['Experience'].sum())  # Total experience by
department
#print("\nAverage Salary by Department:\n",
df.groupby('Department')['Salary'].mean())

# Multiple aggregation
agg_data = grouped.agg({'Salary': ['mean', 'max'], 'Experience':
'sum'})
#print("\nAggregated Data:\n", agg_data)

summary = grouped.agg({
    'Salary': ['mean', 'max', 'min'],
    'Experience': ['mean', 'count']})

#print(summary)
#print(summary.sort_values(('Salary','mean'),ascending=False))
```

```
#print(grouped['Salary'].mean().sort_values(ascending=False))

# Custom function
def salary_range(x):
    return x.max() - x.min()

#print("\nSalary Range by Department:\n",
df.groupby('Department')['Salary'].apply(salary_range))

# Pivot table
pivot = df.pivot_table(values='Salary', index='Department',
aggfunc=['mean', 'max','min'])
print("\nPivot Table:\n", pivot)


# DataFrames for merge
df1 = pd.DataFrame({'EmpID': [1, 2, 3, 4],
                    'Name': ['Aishwarya', 'Bhushan', 'Chetan',
'Dhananjay'],
                    'Department': ['HR', 'Finance', 'IT', 'IT']})

df2 = pd.DataFrame({'EmpID': [3, 4, 5],
                    'Salary': [70000, 80000, 60000]})

print("Left DataFrame:\n", df1)
print("\nRight DataFrame:\n", df2)

# Inner Join
inner_join = pd.merge(df1, df2, on='EmpID', how='inner')
print("\nInner Join:\n", inner_join)

# Left Join
left_join = pd.merge(df1, df2, on='EmpID', how='left')
print("\nLeft Join:\n", left_join)

# Outer Join
outer_join = pd.merge(df1, df2, on='EmpID', how='outer')
print("\nOuter Join:\n", outer_join)

# Concatenation example
df3 = pd.DataFrame({'EmpID': [6, 7],
                    'Name': ['Eknath', 'Faiz'],
                    'Department': ['HR', 'Finance']})

concat_df = pd.concat([df1, df3], ignore_index=True)
print("\nConcatenated DataFrame:\n", concat_df)
```