Minimum Level of Learning Class XII Informatics Practices

तत् त्वं पूषन् अपावृणु
केन्द्रीय विद्यालय संगठन

“SESSION 2025-26

KENDRIYA VIDYALAYA SANGATHAN

Zonal Institute of Education & Training, Gwalior

OUR MENTOR:

MR. B.L. MORODIA

Deputy Commissioner & Director

KVS Zonal Institute of Education and Training, Gwalior

Course Coordinator

MRS. ANITA KANAJIA, T.A. ECONOMICS

RESOURCE PERSONS

1. **MRS SANGEETA M CHAUHAN ,**
   PGT CS , PM SHRI K V NO3 GWALIOR
2. **MR. ALOK GUPTA**
   PGT COMP, PM SHRI K V ETAWAH

VETTED BY:

1. **MR RAJU DIXIT ,**
   PGT CS, PM SHRI K V ALIGARH
2. **MR. RAKESH KUMAR SINGH YADAV**
   PGT CS, PM SHRI K V DOGRA LINES MEERUT CANTT
3. **MR. SUNIL KUMAR BHELE**
   PGT CS, PM SHRI K V PUNJAB LINES MEERUT CANTT
4. **MR. MANISH GUPTA**
   PGT CS, PM SHRI K V HATHRAS
5. **MR PANKAJ SINGH**
   PGT(CS) , PM SHRI KV TALBEHAT

SERIES

Pandas provides the Series data structure, which is a one-dimensional labeled array capable of holding data of any type (integers, strings, floats, etc.). A Series can be created from various data sources, including ndarrays, dictionaries, and scalar values.

1. Creating Series from ndarray An ndarray (NumPy array) can be used to create a Series. If no index is provided, the default index will be [0, 1, 2, ..., n-1].

```
import pandas as pd
import numpy as np
arr = np.array([10, 20, 30, 40])
s = pd.Series(arr)
print(s)
```

```
Output-
            0     10
            1     20
            2     30
            3
40dtype: int64
```

2. Creating Series from Dictionary

A dictionary can be directly converted into a Series where the keys become the index labels, and the values become the data.

```
import pandas as pd
d = {'A': 45, 'B': 67, 'C': 34, 'D': 78}
s = pd.Series(d)
print(s)
```

```
Output-
            A   45
            B   67
            C   34
            D   78
            dtype: int64
```

3. Creating Series from Scalar Value

A scalar value (constant) can be used to create a Series where the value is repeated for all specified indexes.

```
import pandas as pd
s1 = pd.Series(8)
# Creates a Series with one default index (0)
print(s1)
s2 = pd.Series(5, index=['a', 'b', 'c'])
# Repeats 5 for each index
 print(s2)
```

```
Output-
            0 8
            dtype: int64
            a 5
            b 5
            c 5
            dtype: int64
```

Arithmetic Operations in Pandas Series Students will be able to perform and understand basic arithmetic operations (addition, subtraction, multiplication, division) on Pandas Series.

1. Series Addition (+)

Description: Adds values in two Series element-wise (position-wise). If index labels match, values are added; if not, result is NaN.

Code Example:

```
import pandas as pd
s1 = pd.Series([10, 20, 30])
s2 = pd.Series([1, 2, 3])
print(s1 + s2)
```

```
Output-
        0       11
        1       22
        2       33
        dtype:
```

2. Series Subtraction (-)

Description: Subtracts the values of one Series from another element-wise.

Code Example:

```
 print(s1 - s2)
```

Output-

```
0       9
1       18
2       27
dtype: int64
```

3. Series Multiplication (*)

Description: Multiplies values of two Series element-wise.

Code Example:

```
print(s1 * s2)
```

Output-

```
0        10
1        40
2        90
dtype: int64
```

4. Series Division (/)

Description: Divides values of one Series by another element-wise. The result will be in float format.

Code Example:

```
print(s1 / s2)
```

Output-

```
0      10.0
1      10.0
2      10.0
dtype: float64
```

5. Handling Mismatched Indexes

Description: If the Series have different or non-matching indexes, Pandas aligns by index labels. Non-overlapping labels result in NaN.

Code Example:

```
s3 = pd.Series([100, 200], index=['a', 'b'])
s4 = pd.Series([10, 20], index=['b', 'c'])
print(s3 + s4)
```

Output-

```
a    NaN
b    210.0
c    NaN
dtype: float64
```

Key Takeaways:

- *Operations are vectorized (fast and automatic).*
- *Results are aligned by index, not just position.*
- *Missing values are shown as NaN (Not a Number).*
- *Use .fillna() to handle NaNs if needed.*

Head() and Tail() functions in Series

head()	**tail()**
Returns elements from beginning of series	Returns elements from end of series
head(n) Returns the **first** n elements of a Series.	tail(n) returns the **last** n elements of a Series.
if no argument is given, then both functions return 5 elements i.e. head() will give 5 elements from beginning and tail will give 5 elements from end of series	
head(3) ↓↓↓ Index: [0] [1] [2] Data: | 10 | 20 | 30 | ... (rest omitted)	tail(3) ↑↑↑ Data: ... (omitted) | 80 | 90 | 100 | Index: [7] [8] [9]

Selection, Indexing and Slicing in Series

Positional Indexing (Integer-based)	**Label-based Indexing (Custom Index)**
● Starts from 0. ● **Syntax:** seriesname[pos]	● Uses custom-defined labels (if provided). ● **Syntax:** seriesname['label']
print(s[0]) # First element → 10 *print(s[3]) # Fourth element → 40*	*s = pd.Series([10, 20, 30], index=['a', 'b', 'c'])* *print(s['b']) # Output: 20*
Slicing	
Positional Slicing: using loc s[start:stop:step] (exclusive of stop)	**Label-based Slicing**: **using iloc** s['start':'stop'] (inclusive of stop)
# Positional Slicing (exclusive) *print(s[1:3])* *# Elements at positions 1 and 2 → 20, 30*	*# Label-based Slicing (inclusive)* *print(s['a':'c']) # Elements from 'a' to 'c' → 10, 20, 30*

print(s.loc['b']) # Output: 20
print(s.loc['a':'b'])
# a 10 	
# b 20 	
	print(s.iloc[1])
	# Output: 20 (position 1 = 'b')
	print(s.iloc[-1]) # Output: 40 (last element)

print(s.loc[s > 20]) *# Output:* *# c 30* *# d 40*	*print(s.iloc[1:3])* *# Output:* *# b 20* *# c 30*
print(s.loc[['a', 'd']]) *# Output:* *# a 10* *# d 40*	*print(s.iloc[[0, 3]])* *# Output:* *# a 10* *# d 40*
Conditional Selection ● Filters data based on a condition. ● **Syntax:** seriesname[condition]	*print(s[s > 20]) # Returns elements > 20*

DATA FRAME DataFrame is two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns. Pandas DataFrame consists of three principal components, the data, rows, and columns.

Creation of DataFrame :

There are following ways to create a DataFrame :

Creation of an empty DataFrame:

An empty DataFrame can be created as follows:

Coding:

```
import pandas as df
data=df.DataFrame()
print(data)
```

Creation of DataFrame from List of Dictionaries:

We can create DataFrame from a list of Dictionaries, for example:

Coding:

```
 import pandas as pd
d1={'Name':'Ramesh','Age':20,'Marks':75}
d2={'Name':'Dinesh','Age':22,'Marks':88}
d3={'Name':'Suresh','Age':18,'Marks':90}
df1=pd.DataFrame([d1,d2,d3])
print(df1)
```

Notes:

The dictionary keys are taken as column labels

▪ The values corresponding to each key are taken as data

▪ No of dictionaries= No of rows, As No of dictionaries=3, No of rows=3

▪ No of columns= Total Number of unique keys of all the dictionaries of the list, as all dictionaries have same 3 keys, no of columns=3

Coding:

```
 import pandas as pd
d1={'Name':'Ramesh','Age':20,'Marks':75, ‘Gender’:’Male’}
d2={'Name':'Dinesh','Age':22,'Marks':88}
d3={'Name':'Suresh','Age':18,'Marks':90,’Grade’:’B’}
df1=pd.DataFrame([d1,d2,d3])
print(df1)
```

The dictionary keys are taken as column labels

▪ The values corresponding to each key are taken as data

▪ No of dictionaries= No of rows, As No of dictionaries=3, No of rows=3

▪ No of columns= Total Number of distinct keys of all the dictionaries of the list, as total keys is 5, no of columns=5

▪ NaN (Not a Number) is inserted if a corresponding value for a column is missing (As dictionary d1 has no Grade it has Grade as NaN, dictionary d2 has no Gender , hence it has Gender as NaN and d3 has has no Gender and Grade , hence it has both values as NaN)

Creation of DataFrame from Dictionary of Lists:

DataFrames can also be created from a dictionary of lists.

Coding:

```
import pandas as DF
name=['ramya','ravi','abhinav','priya','akash']
age=[16,17,18,17,16]
gender=['f','m','m','f','m']
marks=[88,34,67,73,45]
d1={'name':name,'age':age,'gender':gender,'marks' :marks}
df1=DF.DataFrame(d1)
 print(df1)
```

Note:

Dictionary keys become column labels by default in a Data Frame, and the lists become the rows

Creation of DataFrame from Series:

DataFrame created from One Series:

Coding:

```
import pandas as pd
s1=pd.Series([300,400,500,600])
 df1=pd.DataFrame(s1)
 print(df1)
```

Note:

DataFrame from One Series: No of rows = No of elements in Series=4 (As s1 has 4 elements) No of columns = one (As single series used)

Creation of DataFrame from Dictionary of Series:

Coding:

```
import pandas as pd
name=pd.Series(['ramya','ravi','abhinav','priya','akash'])
age=pd.Series([16,17,18,17,16])
gender=pd.Series(['f','m','m','f','m'])
marks=pd.Series([88,34,67,73,45]) d1={'name':name,'age':age,'gender':gender,'marks':marks}
df1=pd.DataFrame(d1)
print(df1)
```

DataFrame Display

- Understand how to create a DataFrame using pd.DataFrame().

- Use print(df) or simply df in interactive environments to display the DataFrame.

- Know the functions:

- • df.head() – display the first few rows.
- • df.tail() – display the last few rows.
- • df.info() – summary of columns and data types.
- • df.describe() – basic statistics for numerical columns.

DataFrame Iteration

Learn how to loop through DataFrame rows:

• for index, row in df.iterrows(): – iterate row-wise.

• for row in df.itertuples(): – more memory-efficient row-wise iteration.

- Understand the use case: read row values, apply conditions, etc.

DataFrame Operations on Rows and Columns

Add

- Add Column:

```
  df['NewColumn'] = [val1, val2, ...]
```

- Add Row:

```
  new_row = {'col1': val1, 'col2': val2}
  df = pd.concat([df, pd.DataFrame([new_row])], ignore_index=True)
```

Select

- Select Columns:

```
  df['ColumnName']  # single column
  df[['Col1', 'Col2']]  # multiple columns
```

- Select Rows:

```
  df[df['Age'] > 25]  # condition-based selection
  df.loc[1]  # select row by index
```

Delete

- Delete Column:

```
  df.drop('ColumnName', axis=1, inplace=True)
```

- Delete Row:

```
  df.drop(index, inplace=True)  # e.g., df.drop(0, inplace=True)
```

Rename

- Rename Columns:

```
  df.rename(columns={'OldName': 'NewName'}, inplace=True)
```

Head and Tail functions; Indexing using Labels, Boolean Indexing; slicing

Topic	*Definition*	*Examples*	*Question*	*Solution*
Head Function in DataFrame	Displays the first n rows of a DataFrame (default is 5 rows).	df.head(3) displays the first 3 rows.	What does df.head() do?	It shows the first 5 rows by default.
Tail Function in DataFrame	Displays the last n rows of a DataFrame (default is 5 rows).	df.tail(2) displays the last 2 rows.	What does df.tail(3) output?	Displays the last 3 rows of the DataFrame.
Indexing Using Labels in DataFrame	Accessing rows/columns based on their labels using .loc[].	df.loc['A'] selects the row labeled A.	Which function is used for label-based indexing?	.loc[].
Label-based Row Selection	Selects a row using its label.	df.loc['B'] selects the row labeled B.	What does df.loc['B'] return?	Returns the row labeled B.
Label-based Column Selection	Selects a column using its label.	df.loc[:, 'Name'] selects all rows for the column Name.	How can you select a specific column?	Use .loc[:, 'Column_Name'].
Boolean Indexing in DataFrame	Filters rows based on conditions applied to columns.	df[df['Salary'] > 50000] selects rows where salary is greater than 50000.	Define Boolean indexing in DataFrame.	Filtering rows based on conditions.
Boolean Indexing Example	Filters data using expressions.	df[df['Age'] > 30] filters rows where the age is above 30.	What does df[df['Age'] > 30] do?	Filters rows where the age is greater than 30.
Slicing Rows in DataFrame	Extracts specific rows using row index ranges.	df[2:5] selects rows with indices 2, 3, and 4.	Write the syntax for slicing rows in DataFrame.	df[start:end].
Slicing Columns in DataFrame	Extracts specific columns using .iloc[] or .loc[].	df.iloc[:, 0:2] selects the first two columns.	How can slicing simplify data analysis?	By selecting only relevant rows or columns.
Combined Row and Column Slicing	Simultaneous slicing of rows and columns.	df.loc['A':'C', 'Age':'Salary'] selects rows A to C and columns Age to Salary.	What does df.loc['A':'C', 'Age':'Salary'] do?	Selects rows A to C and columns Age to Salary.

DATA VISUALIZATION

Purpose of Plotting

Data visualization helps in understanding trends and patterns by representing data graphically. A **line plot** is used to display changes in data over time or continuous variables.

Line Plot

Matplotlib is a popular Python library for creating graphs.
Here's an example of drawing a simple **line plot**:

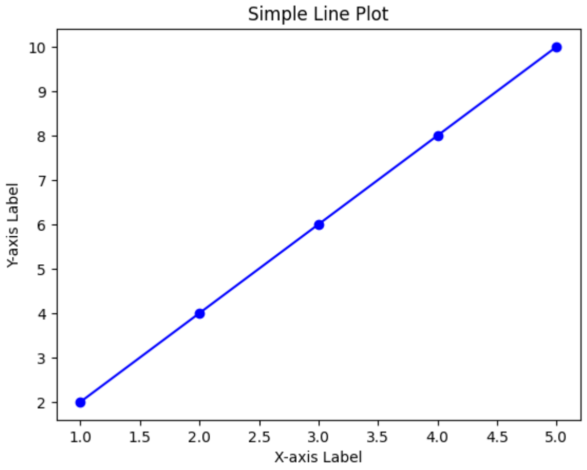
```
import matplotlib.pyplot as plt

# Data points
    x = [1, 2, 3, 4, 5]
    y = [2, 4, 6, 8, 10]

# Creating a line plot
    plt.plot(x, y, marker='o', linestyle='-', color='b')

# Adding labels and title
    plt.xlabel('X-axis Label')
    plt.ylabel('Y-axis Label')
    plt.title('Simple Line Plot')

# Displaying the plot
      plt.show()
```



#Saving the Plot

To save the generated plot as an image file:

```
   plt.savefig('line_plot.png') # Saves the plot as a PNG file
```

This helps in storing the visualization for future reference or reports.

Bar Plot

What is a Bar Plot?

A bar plot is a graph that represents categorical data with rectangular bars. Each bar's length is proportional to the value it represents.

Why Use a Bar Plot?

- To compare different categories or groups.
- To visualize data clearly and quickly.
- Useful for understanding trends and patterns.

Python Library Used

The library used to create bar plots in Python is:
import matplotlib.pyplot as plt

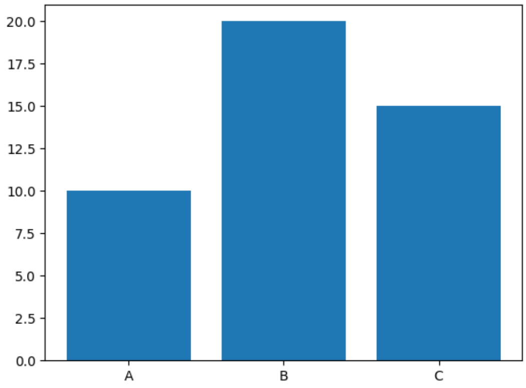
Types of Bar Plots

- Vertical Bar Plot: Uses plt.bar()

Horizontal Bar Plot: Uses plt.barh()

Basic Syntax

```
import matplotlib.pyplot as plt
categories = ['A', 'B', 'C']
values = [10, 20, 15]
plt.bar(categories, values)
plt.show()
```



Customizing a Bar Plot

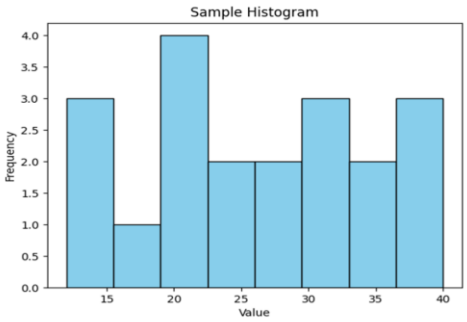
- plt.title('Title') → adds a title
- plt.xlabel('X-axis') → labels x-axis
- plt.ylabel('Y-axis') → labels y-axis
- color='red' → changes bar color
- width=0.5 → changes bar width

Common Functions

- plt.bar()   : Draws vertical bars
- plt.barh()   : Draws horizontal bars
- plt.xlabel()   : Sets x-axis label
- plt.ylabel()   : Sets y-axis label
- plt.title()   : Sets chart title
- plt.xticks()   : Sets labels on x-axis
- plt.savefig() : Saves plot as an image
- plt.show(): Displays the plot

Histogram

Topics	Definition	Example

Histogram	Visualization the number of data points**(Frequency)** that lie within range of values.
	hist() function is used to draw the histogram
		import matplotlib.pyplot as plt
		
		# Sample data
		data = [12, 15, 13, 17, 19, 21, 22, 22, 23, 25, 28, 29, 30, 31, 32, 35, 36, 37, 38, 40]
		
		# Plotting the histogram
		plt.hist(data, bins=8, color='skyblue', edgecolor='black')
		
		# Adding titles and labels
		plt.title('Sample Histogram')
		plt.xlabel('Value')
		plt.ylabel('Frequency')
		
		# Display the plot
		plt.show()
		

grid() function	Configures gridlines in the plot.	
legend()	Display legend of the axis for identification.	
savefig()	Save the plot as image/pdf file type	
title()	Display the title of the plot/graph	
xlabel()	Sets the label of x-axis	
ylabel()	Sets the label of y-axis	
xticks()	Sets the tick location and label on x-axis	
yticks()	Sets the tick location and label on y-axis	

Importing/Exporting Data between CSV files and Dataframes

1.1 Importing data from csv file /Read CSV File to Dataframe/Creating dataframe from csv file

Using the read_csv() function from the pandas package, you can import tabular data from CSV files into pandas dataframe.

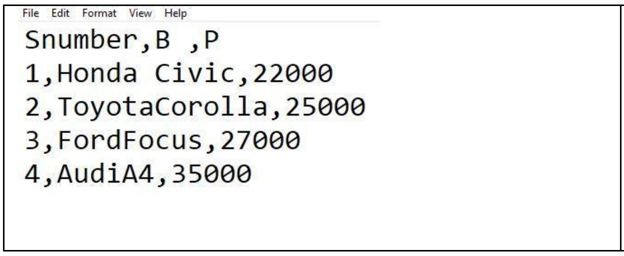
CASE 1. If separator character in csv file is comma

Syntax:

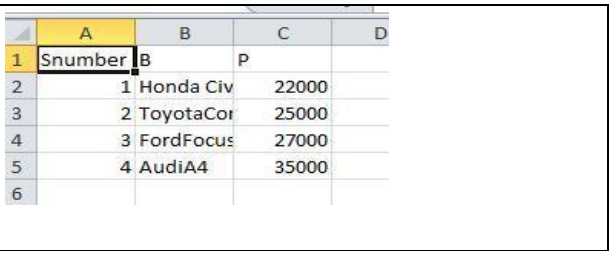
pandas.read_csv(<filepath>)

Eg. Write python code to read a csv file “test.csv” from D:\

Let us assume csv file named test



```
Snumber,B ,P
1,Honda Civic,22000
2,ToyotaCorolla,25000
3,FordFocus,27000
4,AudiA4,35000
```



	A	B	C	D
1	Snumber	B	P	
2	1	Honda Civ	22000	
3	2	ToyotaCor	25000	
4	3	FordFocus	27000	
5	4	AudiA4	35000	
6				

```
>>>import pandas as pd
>>>df = pd.read_csv('d:\\test.csv')                    #read the csv file
>>>df
```

Output:

```
     0    Snumber         B                   P
     1       1           Honda Civic       22000
     2       2           ToyotaCorolla    25000
     3       3           FordFocus         27000
     4       4           AudiA4              35000
```

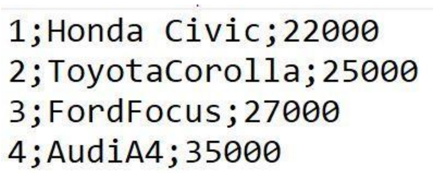
CASE 2. If separator character is different from comma in csv file

Syntax:

<df>=pandas.read_csv(<filepath>,sep=<separator character>)

Eg.

Let us assume csv file named test3



```
1;Honda Civic;22000
2;ToyotaCorolla;25000
3;FordFocus;27000
4;AudiA4;35000
```

```
>>>import pandas
>>>df=pandas.read_csv('d:\\test3.csv',sep=';')
>>>df
     0       1           Honda Civic       22000
     1       2           ToyotaCorolla    25000
     2       3           FordFocus         27000
     3       4           AudiA4              35000
```

1.2 Reading CSV File and specifying own column names

Syntax:

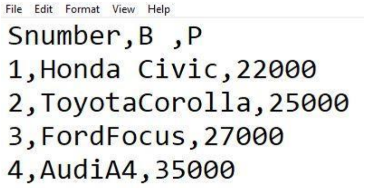
(i) If csv file contains column headings

CASE 1. If we want to change existing column headings

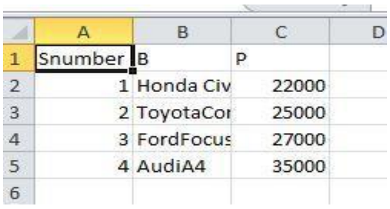
<df>=pandas.read_csv(<filepath>,names=<sequence containing column names>,skiprows=<n>)

In this case, we need to give two arguments along with file path-one for column headings i.e. names=<sequence containing column names> and another for skipping n number of rows while reading data.

Eg. Let us assumn csv file named test.



```
Snumber,B ,P
1,Honda Civic,22000
2,ToyotaCorolla,25000
3,FordFocus,27000
4,AudiA4,35000
```



	A	B	C	D
1	Snumber	B	P	
2	1	Honda Civ	22000	
3	2	ToyotaCor	25000	
4	3	FordFocus	27000	
5	4	AudiA4	35000	
6				

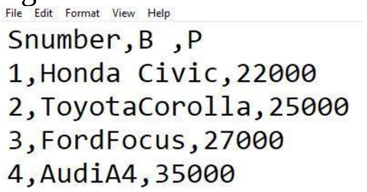
```
>>>import pandas
>>>df=pandas.read_csv(‘d:\\test.csv’,names=[‘Sno’,’Brand’,’Price’],skiprows=1)
>>>df
          Sno           Brand            Price
    0       1           Honda Civic       22000
    1       2           ToyotaCorolla    25000
    2       3           FordFocus         27000
    3       4           AudiA4              35000
```

CASE 2. If we want to give default headings like 0,1,2,3…..

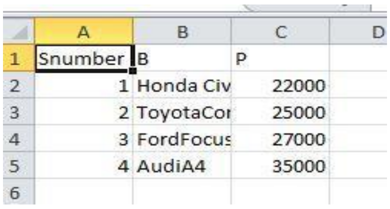
<df>=pandas.read_csv(<filepath>,header=None,skiprows=<n>)

In this case, we need to give two arguments along with file path-one for default column headings i.e. header=None and another for skipping n number of rows while reading data.

Eg. Let us assumn csv file named test.



```
Snumber,B ,P
1,Honda Civic,22000
2,ToyotaCorolla,25000
3,FordFocus,27000
4,AudiA4,35000
```



	A	B	C	D
1	Snumber	B	P	
2	1	Honda Civ	22000	
3	2	ToyotaCor	25000	
4	3	FordFocus	27000	
5	4	AudiA4	35000	
6				

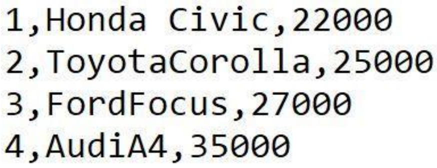
```
>>>import pandas
>>>df=pandas.read_csv(‘d:\\test.csv’,header=None,skiprows=1)
>>>df
            0                1                 2
    0       1           Honda Civic       22000
    1       2           ToyotaCorolla    25000
    2       3           FordFocus         27000
    3       4           AudiA4              35000
```

(ii) If csv file not have column headings

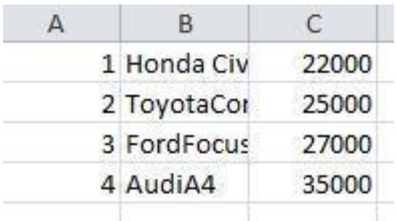
CASE 1. If we want to give column headings

<df>=pandas.read_csv(<filepath>,names=<sequence containing column names>)

Eg. Let us assume csv file names test1



```
1,Honda Civic,22000
2,ToyotaCorolla,25000
3,FordFocus,27000
4,AudiA4,35000
```



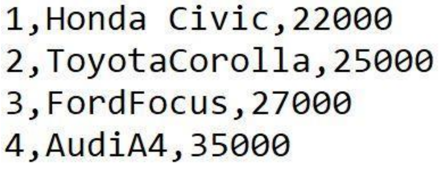
A	B	C
1	Honda Civ	22000
2	ToyotaCor	25000
3	FordFocus	27000
4	AudiA4	35000

```
>>>import pandas
>>>df=pandas.read_csv(‘d:\\test1.csv’,names=[‘Sno’,’Brand’,’Price’])
>>>df
          Sno           Brand            Price
    0       1           Honda Civic       22000
```

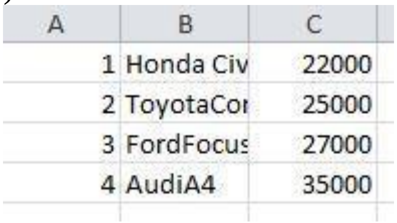
```
      1        2              ToyotaCorolla    25000
      2        3              FordFocus          27000
      3        4              AudiA4               35000
```

CASE 2. If we want to give default headings like 0,1,2,3…..

<df>=pandas.read_csv(<filepath>,header=None)



```
1,Honda Civic,22000
2,ToyotaCorolla,25000
3,FordFocus,27000
4,AudiA4,35000
```



A	B	C
1	Honda Civ	22000
2	ToyotaCor	25000
3	FordFocus	27000
4	AudiA4	35000

```
>>>import pandas
>>>df=pandas.read_csv('d:\\test1.csv',header=None)
>>>df
              0                   1                     2
      0        1              Honda Civic       22000
      1        2              ToyotaCorolla    25000
      2        3              FordFocus          27000
      3        4              AudiA4               35000
```

1.3 Reading specified number of rows from CSV File

Syntax:

<df>=pandas.read_csv(<filepath>,nrows=<n>)

Use nrows=<n> argument for reading n number of rows from csv file to dataframe.

Eg.

```
>>>import pandas
>>>df=pandas.read_csv('d:\\test1.csv',names=['Sno','Brand','Price'],nrows=2)
>>>df
           Sno             Brand              Price
      0       1            Honda Civic       22000
      1       2            ToyotaCorolla    25000
```

1.4 Assigning dataframe index labels from csv file

Syntax:

<df>=pandas.read_csv(<filepath>,index_col=<column_label>)

Use index_col=<column_label> argument for assigning labels of indexes of dataframe from one of the column label of csv file.

Eg.

```
>>>import pandas
>>>df=pandas.read_csv('d:\\test1.csv',names=['Sno','Brand','Price'],index_col="Sno")
>>>df
             Brand                 Price
   Sno
      1       Honda Civic       22000
      2       ToyotaCorolla     25000
      3         FordFocus         27000
      4        AudiA4              35000
```

2.1 Exporting data from dataframe to csv file /Creating csv file from dataframe's data

Using the to_csv() function from the pandas package, you can export data from dataframe to CSV file.

CASE 1. If we want to use comma as a separator character in csv file

Syntax:

<DF>.to_csv(<filepath>)

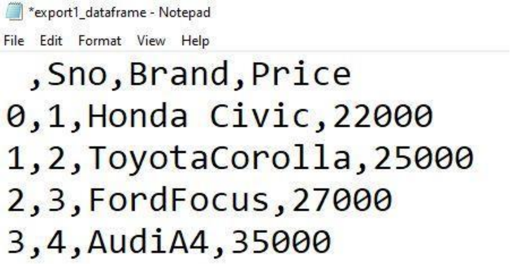
Eg. Write python code to store data present in a dataframe to a csv file "export1_dataframe.csv" in D:\

```
>>>import pandas as pd
>>>cars = {'Brand': ['Honda Civic','ToyotaCorolla','FordFocus','AudiA4'],'Price':
[22000,25000,27000,35000]}
```

```
>>>df= pd.DataFrame(cars, columns= [‘Sno’,'Brand', 'Price'])
>>>df.to_csv(‘d:\\export1_dataframe.csv’)                         #write the csv file
```

Output:

Now open csv file named export1_dataframe in D:/ Drive



```
 ,Sno,Brand,Price
0,1,Honda Civic,22000
1,2,ToyotaCorolla,25000
2,3,FordFocus,27000
3,4,AudiA4,35000
```

CASE 2. If we want to use other separator from comma to separate data in csv file

Syntax:

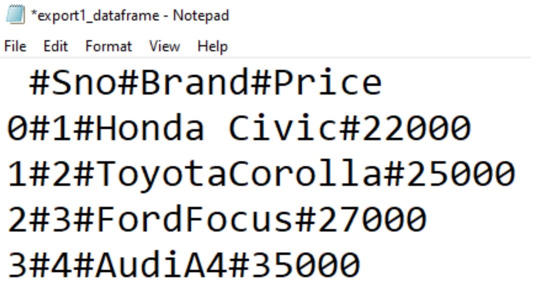
<DF>.to_csv(<filepath>,sep=<separator_character>)

Eg. Write python code to store data present in a dataframe to a csv file “export1_dataframe.csv” in D:\

```
>>>import pandas as pd
>>>cars = {'Brand': ['Honda Civic','ToyotaCorolla','FordFocus','AudiA4'],'Price':
[22000,25000,27000,35000]}
>>>df= pd.DataFrame(cars, columns= [‘Sno’,'Brand', 'Price'])
>>>df.to_csv(‘d:\\export1_dataframe.csv’,sep=’#’)                     #write csv file
```

Output:

Now open csv file named export1_dataframe in D:/ Drive



```
 #Sno#Brand#Price
0#1#Honda Civic#22000
1#2#ToyotaCorolla#25000
2#3#FordFocus#27000
3#4#AudiA4#35000
```

2.2 Skipping row labels/column names in csv file

Syntax:

<DF>.to_csv(<filepath>,header=False,index=False)

Eg. Write python code to store data present in a dataframe to a csv file “export1_dataframe.csv” in D:\

```
>>>import pandas as pd
>>>cars = {'Brand': ['Honda Civic','ToyotaCorolla','FordFocus','AudiA4'],'Price':
[22000,25000,27000,35000]}
>>>df= pd.DataFrame(cars, columns= [‘Sno’,'Brand', 'Price'])
>>>df.to_csv(‘d:\\ export1_dataframe.csv’,sep=’#’,header=False,index=False)                  #write csv
file
```

Output:

Now open csv file named export1_dataframe in D:/ Drive

```
        1#Honda Civic#22000
        2#ToyotaCorolla#25000
        3#FordFocus#27000
        4#AudiA4#35000
```

2.3 Handling NaN values

If dataframe has some missing values, By default, the missing null values are stored as empty strings in csv file.

Eg.

```
>>>df.loc[2,’Brand’]=numpy.NaN
>>>df
          Sno        Brand          Price
    0      1      Honda Civic      22000
    1      2      ToyotaCorolla   25000
    2      3      NaN               27000
```

```
3     4          AudiA4              35000
```

```
>>>df.to_csv('d:\\ export1_dataframe.csv',sep='#',header=False,index=False)
```

Output:

Now open csv file named export1_dataframe in D:/ Drive

```
1#Honda Civic#22000
2#ToyotaCorolla#25000
3#   #27000
4#AudiA4#35000
```

You can specify your own string for missing nan values using argument na_rep=<string>

Eg.

```
>>>df.to_csv('d:\\ export1_dataframe.csv',sep='#',header=False,index=False,na_rep="Null")
```

Output:

Now open csv file named export1_dataframe in D:/ Drive

```
1#Honda Civic#22000
 2#ToyotaCorolla#25000
 3# Null  #27000
 4#AudiA4#35000
```

DATABASE & SQL

Q.NO.	PARTICULARS	MARKS
1.	Write names of two command of DDL & DML	1
2.	Write a command to add a column named remarks with data type varchar size 30 in the table student.	1
3.	Write a function to count number of records in a table employee	1
4.	Find out DDL & DML Commands from the following: INSERT, DELETE, ALTER, DROP	1
5.	Write a query to display all records from the table.	1
6.	What is a primary key?	1
7.	Write one difference between DDL and DML.	1
8.	Write a command to insert a record in the table student with rollno, name, class, marks with values (5,'AMIT','IX',450)	1
9.	Explain following functions avg ( ) and sum( ) with suitable example in sql.	1
10.	Write a command to delete a record having name 'sachin' from the table employee.	1
11.	Write a command to see the structure of the table.	1
12.	Which SQL command is used to retrieve data?	1
13.	Write a command used to arrange the records in descending order in SQL?	1
14.	What is the function of the PRIMARY KEY?	1
15.	What do you mean by Foreign key?	1
16.	Which command is used to change/modify the values in a table.	1
17.	Write the SQL command to display all records from a table named student	1
18.	Aman wants to remove the table Product from the database SHOP which command will he use from the following.	1

19. Write MySQL statements for the following: 2

i. To create a database named FOOD

ii. To create a table named Nutrients based on the following specification:

Column Name	Data Type	Constraints
Food_Item	Varchar(20)	Primary Key
Calorie	Integer	

20. . Consider the following table stored in a database SHOP: 2

Table: Product

Pcode	Pname	Qty	Price
100	Tooth Paste	100	78.0
101	Soap	500	20
102	Talc Powder	50	45.0

(i) What is the degree and cardinality of the above table?

(ii) Write a SQL command to add a new column supcode of char (20) size in the table.

NETWORKING

Q1. Explain different types of computer network
Ans.

● **Type**	● **Full Form**	● **Distance**	● **Media Used**	● **Devices Used**
● PAN	● Personal Area Network	● 30-40 ft ● (A Room)	● Bluetooth, Infrared, Data Cable etc.	●
● LAN	● Local Area Network	● 0-1 Km	● Wifi, Twisted Wire Pair, Ethernet Cable	● Switch/Hub
● MAN	● Metropolitan Area Network	● 1-15Km	● Coaxial Cable, Microwaves	● Repeaters
● WAN	● Wide Area Network	● ∞	● Radio Waves, Optical Fiber, Satellite Communication	● Gateways, Routers

Q2. What do you mean by a hub or a switch?
Ans. Hub: Act as a Central Device in Star Topology. It is a passive device
Switch: is networking hardware that connects devices on a computer network by using packet switching to receive and forward data to the destination device. Also Known as Intelligent Hub.
Q3. What do you mean by a router?

Ans. Router: A router is a networking device that forwards data packets between computer networks. i.e a router connects networks. Routers are intelligent devices, and they store information about the networks they’re connected.

Q4. What do you mean by Network Topology?

Ans. Network Topology: The physical way in which computers are connected to each other in a network is called Network Topology.

Bus: A bus topology is a topology for a Local Area Network (LAN) in which all the nodes are connected to a single cable. The cable to which the nodes connect is called a "backbone". If the backbone is broken, the entire segment fails.

Ring: A ring topology is a network configuration where device connections create a circular data path. Each networked device is connected to two others, like points on a circle.

Star: A star topology is a topology for a Local Area Network (LAN) in which all nodes are individually connected to a central connection point, like a hub or a switch.

Q5. What is URL?

Ans. URL: Uniform Resource Locator : address of a given unique resource on the Web.

e.g. http://www.cbse.nic.in/index.html

Q5. What is VoIP?

Ans. **VoIP**: Voice over Internet Protocol: is a technology that allows you to make voice calls using an Internet connection instead of a regular (or analog) phone line.

Q6. What is the difference between a webpage and a website?

Ans.

● **Webpage**	● **Website**
● Webpage is a single document on the Internet	● Website is a collection of multiple webpages with information on a related topic
● Each webpage has a unique URL.	● Each website has a unique Domain Name

Q7. Write difference between static and dynamic webpage.

Ans. **Static Web Page:** A static web page (sometimes called a flat page or a stationary page) is a web page that is delivered to the user's web browser exactly as stored. i.e. static Web pages contain the same prebuilt content each time the page is loaded

Dynamic web page: The contents of Dynamic web page are constructed with the help of a program. They may change each time a user visit the page. Example a webpage showing score of a Live Cricket Match.

Q8. What do you mean by a web browser? Give Example.

Ans. **Web Browser:** A web browser (commonly referred to as a browser) is a software application for accessing information on the World Wide Web. e.g. Internet Explorer, Google Chrome, Mozilla Firefox, MS Edge, Brave, and Apple Safari.

Q9. Write Full forms of the following:

Ans.	ARPANET	Advanced Research Project Agency Network
	TCP/IP	Transmission Control Protocol / Internet Protocol
	PPP	Point To Point Ptotocol
	VoIP	Voice Over Internet Protocol
	SLIP	Serial Link Internet Protocol
	IMAP	Internet Message Access Protocol
	POP	Post Office Protocol
	PAN	Personal Area Network
	LAN	Local Area Network
	MAN	Metropolitan Area Network
	WAN	Wide Area Network
	MODEM	MOdulator DEModulator
	SIM	Subscriber Identification Module
	Wi-Fi	Wireless Fidelity

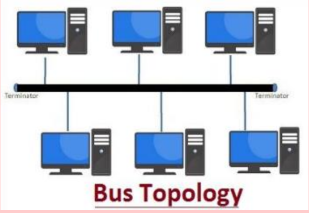
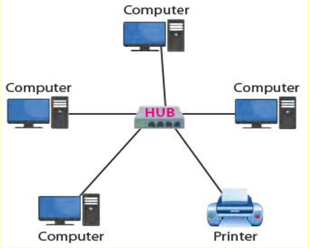
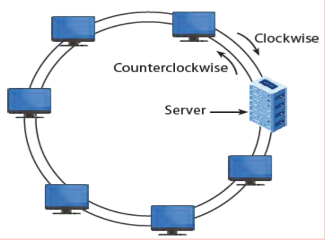
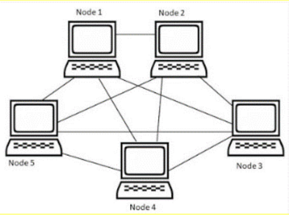
Q10. Explain different types of networks.
Ans.

TYPES OF NETWORKS – Based on geographical area and data transfer rate

PAN (Personal Area Network)	LAN (Local Area Network)	MAN (Metropolitan Area Network)	WAN (Wide Area Network)
Interconnecting few personal devices like laptop, mobile etc. Area – 10 meters Bluetooth / USB	Connects devices in limited area, say office, university campus etc. Area – upto 1 Km Ethernet Cable, Fibre Optics, Wi-Fi etc	Extended form of LAN, within the city. Example – CableTV network in a town. Area – 30-40 Km	Connects devices, LANs and WANs across different parts of country or different countries or continents. Example – Internet

Q11. Explain different kind of topologies.
Ans.

NETWORK TOPOLOGIES - pattern of layout or inter-connection between devices (computer nodes , printers etc.) in a network.

	 Computer Computer Computer HUB Computer Printer		
BUS topology Easy to setup ; Less cable length ; Fault diagnosis difficult; Not suitable for large networks	**STAR topology** Centrally controlled ; Fault diagnosis easy; Expensive to setup; If central hub fails, network disrupts.	**RING topology** Easy to setup ; Higher rate of data transmission; Troubleshooting difficult ;	**MESH topology** Network can be expanded without affecting existing LAN ; Robust topology; Complex setup

Q12. Explain different types of Transmission Media.
Ans.

TRANSMISSION MEDIA	
WIRED (Guided) **Twisted Pair Cable** (Ethernet Cable) Economical and Easy to use stp (shielded twisted pair) , utp (un- shielded twisted pair)	**WIRELESS (Unguided)** **Infrared** – Are electromagnetic radiation for line-of-sight; Frequency 300 GHz - 400 THz; Range 10-30 meters **Bluetooth** - standard wireless (radio wave) communication protocol uses 2.4 GHz frequency; max range 100 meter
Co-axial Cable Example = cable TV wire	**Radio wave** (frequency range 30 Hz – 300 GHz)
Optical Fiber Cable Most reliable, fast transmission, expensive	**Satellite** (downlink frequency 1.5 – 20 GHz) (Uplink frequency 1.6 GHz – 30 GHz) VERY FAST, EXPENSIVE
	Microwave (frequency range 300 MHz – 300 GHz)
All unguided media = transmitter, receiver and atmosphere	

Q13. Explain difference between Router and Bridge.
Ans. ROUTER: It connects multiple networks with different protocols and can handle multiple protocols and works using IP addresses
BRIDGE: connects local networks with same standard but having different types of cables and cannot manage multiple protocols and works using MAC addresses.

Q14. What is a repeater?
Ans. REPEATER is used to re-generate received signal and re-transmit towards destination.

TIP - When to suggest use of Repeater?

When distance between devices is more than 90 meter

Q15. Write difference between a switch and a hub.

Ans.

SWITCH v/s	HUB
An intelligent device that connects several nodes for form a network.	An electronic device which connects several nodes to form a network.
Sends information only to intended nodes	Redirects the information to all the nodes in broadcast form.

Q16. Write tips for case study based QA.

Ans.

Tips for CASE STUDY BASED questions

Question	Hint for Answering
Layout	Draw block diagram interconnecting blocks, prefer the block or unit with maximum devices as main to connect other blocks
Topology	Write name of topology – Star / Bus / Ring etc.
Placement of Server	In the unit/block with maximum number of computers
Placement of Hub/Switch	In every block / unit
Placement of Repeater	As per layout diagram, if distance between two blocks is above 90 meter
Cost-effective medium for internet	Broadband / connection over telephone lines
Communication media for LAN	Ethernet (upto 100 meter) / Co-axial cable for high speed within LAN
Cost/Budget NOT an issue in LAN	Optical Fiber
Communication media for Hills	Radio wave / Microwave
Communication media for Desert	Radio wave
Very fast communication between two cities / countries	Satellite (avoid it in case economical / budget is mentioned)
Device / software to prevent unauthorized access	Firewall (Hardware and/or Software)

Q17. Write difference between http and https.

Ans. HTTP: Hyper Text Transfer Protocol- transfer data from one device to another on the world wide web. HTTP defines how the data is formatted and transmitted over the network.

HTTPS: Hypertext Transfer Protocol Secure: advanced and secure version of HTTP.

Q18. What is an email?

Ans. e-Mail or email, short for "electronic mail," is the transmission of messages electronically over computer networks.

Q19. What are cookies?

Ans. Cookies are combination of data and short codes, which help in viewing a webpage properly in an easy and fast way. Cookies are downloaded into our system, when we first open a site using cookies and then they are stored in our computer only. Next time when we visit the website, instead of downloading the cookies, locally stored cookies are used. Though cookies are very helpful but they can be dangerous, if miss-utilized.

Q20. What are Protocols?

Ans. Protocols are set of rules, which governs a Network communication. Or set of rules that determine how data is transmitted between different devices in a network.

Q21. Write some advantages of Computer Network.

Ans. Some advantages are as follows:

- We can share resources such as printer and scanner
- Can share data and access files from any machine
- Software can be installed on server and used on client machine
- Save cost

E-Waste and Its Management —
Definition of E-Waste

- **E-Waste** or **Electronic Waste** includes discarded electronic devices like computers, mobile phones, TVs, printers, etc., that have reached the end of their useful life.

Why E-Waste is a Problem

- E-waste is growing rapidly due to increased use of electronics.
- Lack of awareness and skills in disposal worsen the issue.
- It makes up more than **5% of municipal solid waste** globally.

Impact of E-Waste on Environment

- Improper disposal causes **air, water, and soil pollution**.
- **Toxic metals** leach into soil and groundwater.
- **Burning e-waste** releases harmful gases into the air.

Impact of E-Waste on Humans

- **Lead**: Causes lead poisoning, affects brain and kidneys.
- **Beryllium**: From burnt circuit boards; causes skin diseases and lung cancer.
- **Mercury**: Affects the respiratory system and brain.
- **Cadmium**: Damages liver, kidneys, and bones.
- **Plastics**: Release chemicals that harm the immune system and mental health.

E-Waste Management - 3 R’s

1. **Reduce**: Buy electronics only when needed, use them fully.
2. **Reuse**: Donate or sell working devices; called **refurbishing**.
3. **Recycle**: Recover materials from non-repairable devices.

IN E-Waste Management in India

- **Environmental Protection Act, 1986**: "Polluter Pays Principle" – the polluter must pay for environmental damage.
- **CPCB Guidelines**: Manufacturers are responsible for safe disposal.
- **DIT Guidelines**: Reuse and recycling encouraged through technical manuals.
- Companies run **e-waste recycling programs** to collect devices.

Health Impact of Digital Devices

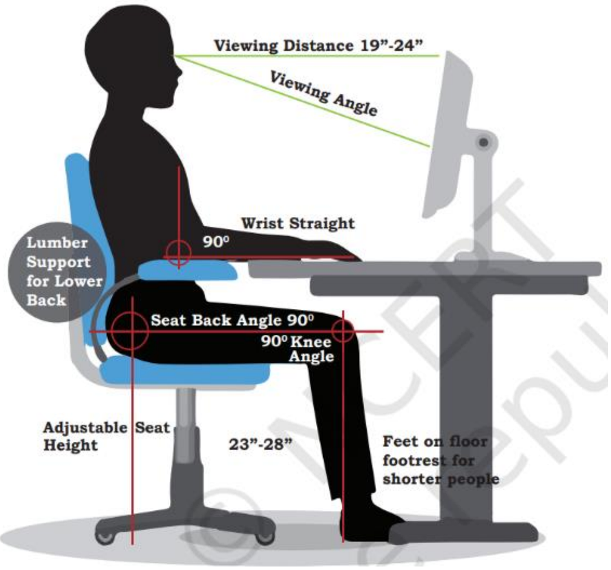
- Excessive screen time causes **eye strain, stress, obesity**, etc.
- Poor posture causes **back, neck, and wrist problems**.
- Prolonged use of devices can cause **physical and psychological issues**.

What is Ergonomics?

- **Ergonomics** is the science of designing or arranging workplaces, furniture, and devices to make them **safe, efficient, and comfortable** for users.
- It helps reduce **physical strain, fatigue, and injuries** caused by prolonged use of digital devices.

Ergonomic Guidelines for Computer Usage:

- **Monitor Position**:
  - The **top of the screen** should be at or just **below eye level**.
  - The screen should be at a **distance of 19–24 inches** from your eyes.
  - The **viewing angle** should be around **15° to 20° downward** from the horizontal eye level.
  - **Chair and Posture**: Feet should rest **flat on the floor** or on a footrest.
  - Knees should be at a **90° angle**.
  - Back should be straight and supported by the **chair’s backrest**.
- **Keyboard and Mouse**:
  - Should be placed at **elbow height**.
  - Elbows should form a **90° angle**.
  - Wrists should be in a **neutral (straight)** position while typing.
- **Breaks and Eye Care**:
  - Follow the **20-20-20 rule**: Every 20 minutes, look at something 20 feet away for 20 seconds.
- Blink frequently to prevent dry eyes
  - .



Societal impacts

One of the most significant outcomes of the progress of information technology is probably electronic commerce over the Internet, a new way of conducting business. Though only a few years old, it may radically alter economic activities and the social environment. Already, it affects such large sectors as communications, finance and retail trade and might expand to areas such as education and health services. It implies the seamless application of information and communication technology along the entire value chain of a business that is conducted electronically. The following sections will focus on the impacts of information technology and electronic commerce on business models, commerce, market structure, workplace, labour market, education, private life and society as a whole.

Digital Footprint

A digital footprint is data that is left behind when users have been online. There are two types of digital footprints which are passive and active.

A passive footprint is made when information is collected from the user without the person knowing this is happening.

An active digital footprint is where the user has deliberately shared information about themselves either by using social media sites or by using websites.

An example of a passive digital footprint would be where a user has been online and information has been stored on an online database. This can include where they came from, when the footprint was created and a user IP address. A footprint can also be analysed offline and can be stored in files which an administrator can access. These would include information on what that machine might have been used for, but not who had performed the actions.

An example of an active digital footprint is where a user might have logged into a site when editing or making comments such as on an online forum or a social media site. The registered name or profile can be linked to the posts that have been made and it is surprisingly easy to find out a lot about a person from the trails you leave behind.

Net and Communication Etiquettes

1. Be respectful.
2. Be aware of how your comments might be read:
3. Be careful with humour and sarcasm
4. Think about who can see what you have shared.
5. Remember to check friend requests and group invites before accepting them.
6. Take time to have a read of the rules of conduct/ community standards.
7. Be forgiving.

Data Protection

Data protection is a set of strategies and processes you can use to secure the privacy, availability, and integrity of your data. It is sometimes also called data security or information privacy. A data protection strategy is vital for any organization that collects, handles, or stores sensitive data.

Data Protection vs Data Privacy

Although both data protection and privacy are important and the two often come together, these terms do not represent the same thing.

One addresses policies, the other mechanisms

Data privacy is focused on defining who has access to data while data protection focuses on applying those restrictions. Data privacy defines the policies that data protection tools and processes employ.

Creating data privacy guidelines does not ensure that unauthorized users don’t have access. Likewise, you can restrict access with data protections while still leaving sensitive data vulnerable. Both are needed to ensure that data remains secure.

Another important distinction between privacy and protection is who is typically in control. For privacy, users can often control how much of their data is shared and with whom. For protection, it is up to the companies handling data to ensure that it remains private. Compliance regulations reflect this difference and are created to help ensure that users’ privacy requests are enacted by companies.

Data Protection Technologies and Practices that Can Help You Protect User Data

When it comes to protecting your data, there are many storage and management options you can choose from. Solutions can help you restrict access, monitor activity, and respond to threats. Here are some of the most commonly used practices and technologies:

1. **Data loss prevention (DLP)—**a set of strategies and tools that you can use to prevent data from being stolen, lost, or accidentally deleted. Data loss prevention solutions often include several tools to protect against and recover from data loss.

2. **Storage with built-in data protection**—modern storage equipment provides built-in disk clustering and redundancy. For example, Cloudian's Hyperstore provides up to 14 nines of durability, low cost enabling storage of large volumes of data, and fast access for minimal RTO / RPO.
3. **Firewalls**—utilities that enable you to monitor and filter network traffic. You can use firewalls to ensure that only authorized users are allowed to access or transfer data.
4. **Authentication and authorization**—controls that help you verify credentials and assure that user privileges are applied correctly. These measures are typically used as part of an identity and access management (IAM) solution and in combination with role-based access controls (RBAC).
5. **Encryption**—alters data content according to an algorithm that can only be reversed with the right encryption key. Encryption protects your data from unauthorized access even if data is stolen by making it unreadable. Learn more in our article: Data Encryption: An Introduction.
6. **Endpoint protection—**protects gateways to your network, including ports, routers, and connected devices. Endpoint protection software typically enables you to monitor your network perimeter and to filter traffic as needed.
7. **Data erasure**—limits liability by deleting data that is no longer needed. This can be done after data is processed and analyzed or periodically when data is no longer relevant. Erasing unnecessary data is a requirement of many compliance regulations, such as GDPR. For more information about GDPR, check out our guide: GDPR Data Protection.

<u>Intellectual Property Rights</u>

Property

The word property is generally used to mean a possession or, more specifically, something to which the owner has legal rights

Intellectual Property

It refers to creations of the intellect used in commerce:

- ➢ Inventions
- ➢ Literary and Artistic work
- ➢ Symbols
- ➢ Names
- ➢ Images and designs

		Interventions (patent)
	INDUSTRIAL PROPERTY	Trademarks
		Commercial names
		Location Specific brands
		Literary Works
		Novels
		Poems
		Plays
Categories of Intellectual Property: -		Film & Media Works
		Artistic Works
	COPYRIGHTS	Drawings
		Paintings
		Photographs
		Sculptures
		➢ Architectural designs - Images and designs

Copyright laws protect Intellectual property

Copyright

It is a legal concept , enacted by most governments giving creator of original work exclusive rights to it, usually for a limited period.

Plagiarism

It is stealing someone's intellectual work and representing it as your own work without citing the source of information.

Copying someone's work and then passing it off as one's own

- Act of stealing
- Copying information and not giving the author credit for it
- Copying programs written by other programmers and claiming them as your own
- Involves lying, cheating, theft and dishonesty

Measure to avoid Plagiarism:

- Plagiarism is a bad practice and should be avoided by the following measures:
- Use your own words and ideas.
- Always provide reference or give credit to the source from where you have received your information.
- You must give credit whenever you use
- Another person's idea, opinion, or theory.
- Quotations of another person's actual spoken or written words
- Paraphrase of another person's spoken or written words.

Licensing:

Software Licensing is the legal right to run or the privilege gives to you by a company to access their application or program or software.

For example:

When we purchase for proprietary software such as Windows OS, then we must have noticed that it comes with a license agreement which is to be read first and to be agreed upon for the successful installation and usage of the software.

License agreements typically allow the software to run on a limited number of computers and allow copies to be made only for backup purpose.

Advantages of using Licensed software

1. By using licensed software, you are able to contribute to the further development of the program you are using.

2. It comes with the outright support not found in "pirated" software.

Free and open-source software

Free and open-source software (FOSS) is software that can be classified as both free software and open-source software. That is, anyone is freely licensed to use, copy, study, and change the software in any way, and the source code is openly shared so that people are encouraged to voluntarily improve the design of the software. This is in contrast to proprietary software, where the software is under restrictive copyright licensing and the source code is usually hidden from the users.

Free software

Free Software Foundation (FSF), defines free software as a matter of liberty not price, and it upholds the Four Essential Freedoms.

Four essential freedoms of Free Software

- The freedom to run the program as you wishes, for any purpose (freedom 0).
- The freedom to study how the program works and change it so it does your computing as you wish (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help others (freedom 2).
- The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.

Open Source Software

Open-source software is computer software that is released under a license in which the copyright holder grants users the rights to use, study, change, and distribute the software and its source code to anyone and for any purpose. Open-source software may be developed in a collaborative public manner.

***** Hard Work Never Goes in Vein*****