# Balram Mandal

# Roll No : 30

EXP.NO.12

Program to demonstrate DataFrame using Pandas

import pandas as pd

Creating a DataFrame using a dictionary

|  |
| --- |
| data = {  'Name': ['Alice', 'Bob', 'Charlie', 'David'],  'Age': [24, 27, 22, 32],  'Marks': [85, 78, 90, 88]  }  df = pd.DataFrame(data) print("Initial DataFrame:\n", df)  Initial DataFrame:  Name Age Marks   1. Alice 24 85 2. Bob 27 78 3. Charlie 22 90 4. David 32 88 |

Accessing columns

print

(

"Accessing 'Name' column:

\n

"

, df[

'Name'

])

Accessing 'Name' column

:

0 Alic

e

1

b

Bo

e

Charli

2

d

Davi

3

dtype: object

Name: Name,

Adding a new column

df[

'Grade'

]

=

[

'B'

,

'C'

,

'A'

,

'A'

]

print

(

"DataFrame after adding 'Grade' column:

\n

"

, df)

DataFrame after adding 'Grade' column:

Name Age Marks Grade

0

B

Alice 24 85

1

C

Bob 27 78

2

A

Charlie 22 90

3

David 32 88

A

Accessing a specific row using loc

print("Accessing row at index 2:\n", df.loc[2])

Accessing row at index 2:

Name Charlie

Age 22

Marks 90

Grade A

Name: 2, dtype: object

Accessing multiple rows using iloc

print("Accessing rows from index 1 to 3:\n", df.iloc[1:4])

Accessing rows from index 1 to 3:

Name Age Marks Grade

1. Bob 27 78 C
2. Charlie 22 90 A
3. David 32 88 A

Filtering data

high\_scorers = df[df['Marks'] > 85]

print("Students scoring above 85 marks:\n", high\_scorers)

Students scoring above 85 marks:

Name Age Marks Grade

1. Charlie 22 90 A
2. David 32 88 A

Basic statistics

print("Statistical summary:\n", df.describe())

Statistical summary:

Age Marks count 4.000000 4.000000 mean 26.250000 85.250000 std 4.349329 5.251984 min 22.000000 78.000000 25% 23.500000 83.250000

50% 25.500000 86.500000 75% 28.250000 88.500000 max 32.000000 90.000000