

Directorate of Education, GNCT of Delhi
Mid Term Examination Practice Paper
Session: 2025-26
CLASS-XII
Informatics Practices (065)

Time allowed: 3 Hours

Maximum Marks: 70

General Instructions:

1. All questions are compulsory.
2. This question paper contains five sections: A, B, C, D, and E.
3. Section A has 21 questions (1 mark each).
4. Section B has 7 questions (2 marks each).
5. Section C has 4 questions (3 marks each).
6. Section D has 2 questions (4 marks each).
7. Section E has 3 questions (5 marks each).
8. All programming questions are to be answered using Python Language only.

Section A (1 mark each)

1. True/False: The FOREIGN KEY constraint ensures uniqueness of values in a column.
2. Which of the following can be used to create a Pandas DataFrame?
 - a) A dictionary of lists
 - b) A dictionary of NumPy arrays
 - c) A single scalar value without index
 - d) A list of Series
3. The tail(4) function in Pandas returns:
 - a) First 4 rows
 - b) Last 4 rows
 - c) First 4 columns
 - d) Random 4 rows
4. Which SQL function removes spaces from both ends?
 - a) TRIM()
 - b) ROUND()
 - c) MID()
 - d) MOD()
5. To know dimensions of DataFrame df:
 - a) df.shape

- b) df.ndim
 - c) df.size
 - d) df.axes
6. Which SQL function converts to lowercase?
- a) LCASE()
 - b) UPPER()
 - c) LENGTH()
 - d) LEFT()
7. Correct way to import matplotlib?
- a) import matplotlib.pyplot as plt
 - b) import matplotlib as plt
 - c) include matplotlib
 - d) import pyplot as plt
8. Select multiple columns Marks, Grade from df:
- a) df[['Marks','Grade']]
 - b) df.get('Marks','Grade')
 - c) df.loc['Marks','Grade']
 - d) df['Marks','Grade']
9. True/False:
xlabel() adds a title to graph.
10. Which accessor retrieves rows by labels?
- a) .iloc[]
 - b) .loc[]
 - c) .ix[]
 - d) .get()
11. Correct command to read CSV 'student.csv':
- a) pd.read_csv('student.csv')
 - b) pd.load_csv('student.csv')
 - c) pd.open_csv('student.csv')
 - d) pd.csv('student.csv')
12. Output of SELECT SUBSTR('Database',2,3);
- a) ata
 - b) tab
 - c) dat
 - d) bas
13. To display x-axis label:
- a) plt.title()
 - b) plt.xlabel()
 - c) plt.ylabel()
 - d) plt.legend()

14. Adding 10 to Series [1,2,3] results in:
- a) Error
 - b) [11,12,13]
 - c) [1,2,3,10]
 - d) Unchanged
15. Command to drop column 'Marks':
- a) `df.drop('Marks',axis=1)`
 - b) `del df.Marks,axis=1`
 - c) `df.remove('Marks')`
 - d) `df.delete('Marks')`
16. `df.loc[df['Price']<500]` is:
- a) Boolean Indexing
 - b) Slicing
 - c) Position Indexing
 - d) Reshaping
17. Output of `SELECT ROUND(78.567,2);`
- a) 78.5
 - b) 78.56
 - c) 78.57
 - d) 79
18. In Pandas, `df.iloc[0:3]` selects:
- a) First 3 columns
 - b) First 3 rows
 - c) Row at index 3
 - d) Column at 0 to 3
19. Correct syntax for Series:
- a) `pd.Series([2,4,6])`
 - b) `pd.CreateSeries([2,4,6])`
 - c) `pd.Series(list=[2,4,6])`
 - d) `Series([2,4,6])`
20. **Assertion(A):** A bar chart is created using `plt.bar()`.
Reason(R): Bar charts compare categorical data.
21. **Assertion(A):** `AVG()` ignores NULL.
Reason(R): Aggregate functions return a single value.

Section B (2 marks each)

22. Difference between line chart and bar chart with example.

OR

Explain purpose of `info()` and `describe()` in Pandas.

23. Difference between head() and tail() in Pandas.

24. Predict output:

```
import pandas as pd
S=pd.Series([3,6,9,12])
print(S.head(3))
print(S.tail(2))
```

25. Differentiate LCASE() and ROUND() with examples.

OR

Differentiate GROUP BY and ORDER BY in SQL with example.

26. Correct the error:

```
import pandas
s = pd.Series([5,10,15])
print(s)
```

27. Output of:

```
import pandas as pd
s=pd.Series([2,4,6,8],index=['w','x','y','z'])
print(s['y'])
```

OR

```
import pandas as pd
s=pd.Series([11,22,33,44])
print(s*3)
```

28. Fill in blanks:

```
import _____ as pd
data=[7,14,21,28]
S=pd._____(data)
print(_____)
```

Section C (3 marks each)

29. SQL command to create CUSTOMER(C_ID,VARCHAR(4),PK; C_Name,VARCHAR(20); City,VARCHAR(15)). Delete record C_ID='C002'.

OR

Create ORDERS(O_ID PK, O_Name, Price). Insert ('O101','Notebook',75).

30. DataFrame:

```
data={'Name':['Amit','Bina','Chetan'],'Age':[17,18,17],'Marks':[85,92,78]}
df=pd.DataFrame(data)
```

- (i) Add column 'Result'='Pass'
- (ii) Rename Marks to Score
- (iii) Delete Age

31. Series:

```
S=pd.Series([15,25,35,45,55])
```

- (i) Show first 3 elements
- (ii) Show elements > 30

OR

```
Marks=pd.Series([90,45,60,80,35])
```

- (i) Show last 2 elements
- (ii) Show elements < 70

32. What is Pandas Series? Write 2 features. Show example of creating Series using dictionary.

Section D (4 marks each)

33. Table Student(Sid,Sname,Marks,Class,DOB). Write SQL queries to show:

- (i) Names with Marks > 75
- (ii) Class-wise average Marks
- (iii) Year of birth of each student
- (iv) Students ending with 'a'

34. Output of:

```
import pandas as pd
S=pd.Series([2,4,6,8,10],index=['p','q','r','s','t'])
print('Part A')
print(S.tail(3))
print('Part B')
print(S['r'])
print('Part C')
print(S<7)
```

OR

```
import pandas as pd
S=pd.Series([50,100,150,200],index=['m','n','o','p'])
print('Part A')
print(S.head(2))
print('Part B')
print(S['n'])
```

```
print('Part C')
print(S>120)
```

Section E (5 marks each)

35. Table LIBRARY(BookID,Title,Author,Price,Copies,PubDate)

BookID	Title	Author	Price	Copies	PubDate
B01	Physics	NCERT	450	20	2023-07-15
B02	Chemistry	TataMcG	500	15	2024-01-20
B03	Maths	NCERT	600	10	2023-12-10
B04	English	Oxford	350	NULL	2024-02-18
B05	CS	Pearson	550	25	2024-03-05

Write SQL query of following statement :

- (i) All details where Author='NCERT'
- (ii) UPPER(Title) where Price>400
- (iii) ROUND(Price,-2) for B02
- (iv) Month of PubDate for Title='CS'
- (v) Total Copies

36. Fill blanks in bar chart:

```
import pandas as pd
import matplotlib.pyplot as plt
data={'Subject':['Math','CS','Physics'],'Marks':[85,92,78]}
df=pd.DataFrame(data)
plt.bar(df['_____'],df['_____']) #(a,b)
plt.title('_____') #(c)
plt.xlabel('_____') #(d)
plt.ylabel('_____') #(e)
plt.show()
```

37. DataFrame:

```
data={'Roll':[11,12,13],'Name':['Rohit','Sneha','Isha'],'Marks':[75,88,95]}
df=pd.DataFrame(data)
(i) Set Roll as index
(ii) Display Roll=13
(iii) Add Roll=14 Name='Kunal' Marks=82
(iv) Display Names>80
(v) Average Marks
```

OR

```
df={'City':['Pune','Delhi','Mumbai'],'Temp':[28,32,30],'Humidity':[55,60,65]}
(i) Set City index
```

- (ii) Humidity of Delhi
- (iii) Add Chennai,34,50
- (iv) Cities Temp>30
- (v) Average Humidity