

Series JPR_PB/25-26/12/065/SET No.8
Class XII –IP
SET-8

Section A (1 mark × 21 = 21 Marks)

Q.No	Question Summary	Answer
1	Matplotlib can generate many types of plots	True
2	SELECT POWER(2, MOD(17, 3));	MOD(17,3)=2 → $2^2=4$ → <input checked="" type="checkbox"/> (d) 4
3	Free software with source code open for modification	(C) Free and Open Source Software
4	CSV stands for	(a) Comma Separated Values
5	Device to amplify signal in Ethernet cable	(D) Repeater
6	Now function	(A) current date and time
7	Protection for a poem	(B) Copyright
8	DataFrame size and values nature	(b) mutable, mutable
9	Add primary key after table creation	(c) ALTER TABLE
10	Application of VoIP	(C) Internet Telephony
11	Count non-NULL values	(B) COUNT(column_name)
12	Create DataFrame from	(d) All of the above
13	Main Indian cyber law	(C) Information Technology Act, 2000
14		(D) order by
15	First 2 rows (0,1) of dfrm	(A) dfrm.loc[:1]
16	Software displaying webpages	(B) Web Browser
17	Use of INSTR()	(C) Finds position of substring
18	Display first 3 rows of Series HEAD	(A) print(HEAD.head(3))
19	NOT aggregate function	(C) UPPER()
20	Assertion–Reason: DataFrame has both row & col index; .loc() selects labels	(A) Both True, R explains A
21	A: CREATE TABLE is DML; R: DDL modifies data	Both false → (D)

Section B (2 marks × 7 = 14 Marks)

22. (B)

DataFrame: 2D labelled data structure with rows & columns.

Property: Both rows and columns have labels (indexes).

OR

Series → 1D; DataFrame → 2D

Series holds single data type; DataFrame may have mixed types.

23.E-waste: discarded electronic devices.

Hazardous substances: **Lead (Pb)** and **Mercury (Hg)** (also Cadmium, Chromium, etc).

24.

```
import pandas as pd
marks_data = {"Hindi":55, "English":65, "Science":35}
s = pd.Series(marks_data)
print(s)
```

25.

Add-ons vs Plugins:

- Add-ons: additional software features integrated into browsers.
- Plugins: external programs that add specific capabilities (e.g., Flash).

OR

Web browser: Software to access/view webpages.

Examples: **Google Chrome, Mozilla Firefox**

26.

I. SELECT YEAR('2025-01-01');

II. SELECT INSTR('Computer Science','Science');

27.

(i) **Cyber Bullying:** Use of electronic means to harass or intimidate someone.

(ii) **Cyber Stalking:** Repeated online harassment or monitoring of someone's activities.

28. (A)

```
import pandas as pd
S1 = pd.Series([1,2,3,4])
S2 = pd.Series([7,8,9,10])
S2.index = ['a','b','c','d']
print((S1+S2).count())
```

→ Different indexes → NaN → Count = **0**

OR (B)

```
import pandas as pd
vaccine_qty = pd.Series([10,16,1], index=["Typhoid","Tetanus","Hepatitis"])
cost = pd.Series([200,500,800], index=["Typhoid","Tetanus","Flu"])
print(vaccine_qty + cost)
```

Output:

Flu

NaN

```
Hepatitis      NaN
Tetanus       516.0
Typhoid        210.0
dtype: float64
```

Section– C (3 marks × 4 = 12 Marks)

29.

I. **Intellectual Property (IP):** Creations of mind like inventions, artistic works, designs, symbols.

IPR: Legal rights given to creators to protect their IP.

II. Seema's invention → **Patent.**

III. IPR encourages innovation, prevents misuse, ensures financial reward.

30.

(A)

```
import pandas as pd
import numpy as np
runs = np.array([88,65,44,27])
players = ['Sachin','Rahul','Yuvraj','Dhoni']
s = pd.Series(runs, index=players)
print(s)
```

OR

```
import pandas as pd
data = [
    {"Course": "DECE", "duration": 24},
    {"Course": "DIPLOMA", "duration": 12},
    {"Course": "DEGREE", "duration": 36},
    {"Course": "CERTIFICATE", "duration": 6}
]
df = pd.DataFrame(data)
print(df)
```

31.

```
(i) CREATE DATABASE TRAVELER;
(ii) CREATE TABLE TRAVELS (
    Tid INT PRIMARY KEY,
    Tname CHAR(20),
    Loc1 CHAR(30),
    Journey_Date DATE,
    Charges DECIMAL(6,2)
);
```

32. (A)(VAHAN table example)

- i. SELECT Year, MAX(Price) FROM VAHAN GROUP BY Year;
- ii. SELECT Company, Year FROM VAHAN ORDER BY Price DESC;
- iii. SELECT Year, SUM(Quantity), COUNT(*) FROM VAHAN GROUP BY Year;

OR

(B)

- i. SELECT COMPANY,COUNT(*) FROM FLIGHTS GROUP BY COMPANY;
 - ii. SELECT UPPER(PASSENGER) FROM BOOKING WHERE DESTINATION="MUM";
 - iii. SELECT PASSENGER,SOURCE,DESTINATION,MODEL FROM FLIGHTS F,BOOKING B WHERE F.FLIGHTID=B.FLIGHTID;
-

Section– D (4 marks × 2 = 8 Marks)

33.

```
import matplotlib.pyplot as plt      # Statement 1
y = ['one', 'two', 'three', 'four', 'five']
x = [5, 24, 35, 67, 12]
plt.barh(y, x)                      # Statement 2
plt.ylabel("pen sold")               # Statement 3
plt.xlabel("price")
plt.title("Horizontal Bar Graph")    # Statement 4
plt.show()
```

34. (A)

a.

```
SELECT Type, AVG(Price)
FROM vehicle
GROUP BY Type
HAVING SUM(Quantity) > 20;
```

b.

```
SELECT Company, COUNT(Type)
FROM vehicle
GROUP BY Company;
```

c.

```
SELECT SUM(Price) FROM vehicle;
```

d.

```
SELECT DISTINCT Type FROM vehicle;
```

OR (B) Output:

- a. VehicleName where Color='WHITE' → **Innova, A-Star**
- b. Order by Capacity → Indigo(3), A-Star(3), SX4(4), C Class(4), Innova(7)
- c. max(Charges)=35, sum(Capacity)=21
- d. Join →

CName	VehicleName
-------	-------------

Raj Lal	Indigo
Feroza Shah	A-Star
Ketan Dhal	C Class

Section – E (5 marks × 3 = 15 Marks)

35. (Networking Scenario)

- I. Layout: Use **star topology** with **ADMIN** as central node (shortest total distance).
 - II. Server at **ADMIN** – it has maximum (120 PCs) & central connectivity.
 - III. No repeater needed — distance < 100m between any pair.
 - IV. Place **switch/hub** in each building; main switch in **ADMIN** for interconnection.
 - V. To connect Mumbai ↔ Jaipur → **WAN** (Wide Area Network).
-

36.

```
import pandas as pd

# Create the DataFrame

df = pd.DataFrame({
```

```
    'Name': ['Ramesh', 'Neeraj', 'Seema', 'Aditya', 'Naresh'],
```

```
    'Unit': ['HR', 'IT', 'Finance', 'IT', 'Marketing'],
```

```
    'Salary': [45000, 33000, 87000, 33000, 25000]
```

```
})
```

```
# I. Print the last three rows of the DataFrame df
```

```
print(df.tail(3))
```

```
# II. Add a new column named "Experience" with values [3, 8, 1, 6, 7]
```

```
df['Experience'] = [3, 8, 1, 6, 7]
```

```
# III. Delete the column "Salary" from the DataFrame
```

```
df = df.drop('Salary', axis=1)
```

```
# IV. Rename the column "Unit" to "Dept"
```

```
df = df.rename(columns={'Unit': 'Dept'})
```

```
# V. Display only the "Name" and "Salary" columns from the DataFrame
```

```
# (Since we deleted 'Salary' earlier, we'll use the original df again for this part)

df_original = pd.DataFrame({  
    'Name': ['Ramesh', 'Neeraj', 'Seema', 'Aditya', 'Naresh'],  
    'Unit': ['HR', 'IT', 'Finance', 'IT', 'Marketing'],  
    'Salary': [45000, 33000, 87000, 33000, 25000]  
})  
  
print(df_original[['Name', 'Salary']])
```

37. (A)

- I. SELECT LEFT(emp_code, 4) FROM Employee;
- II. SELECT COUNT(Order_Id) FROM Orders;
- III. SELECT YEAR(order_date) FROM Orders;
- IV. SELECT TRIM(Address) FROM Customers;
- V. SELECT CURRENT_DATE();

OR (B)

- I. SELECT LENGTH('DatabaseSystems');
 - II. SELECT INSTR(Product_Name, 'a') FROM Products;
 - III. SELECT POW(Tran_Amount, 2) FROM Transactions;
 - IV. SELECT AVG(Salary) FROM Employees;
 - V. SELECT SUM(Salary) FROM Employees;
-