MARKING SCHEME - SAMPLE PAPER – 1
SUBJECT: INFORMATICS PRACTICES

SuperNova-LearnPython, a YouTube channel dedicated to helping students to learn Python and computer science concepts.

The channel covers various topics related to computer science, including ***Python programming, data file handling, computer networking***, ***SQL*** and many more.

If you’re looking for video descriptions, notes, assignments, and previous years’ question papers related to Python and computer science for class 11 and 12, I recommend checking out the **SuperNova-LearnPython** channel on YouTube.

You can find the channel here[1].

Happy learning! .

Please like, Subscribe and share the Channel

SuperNova-LearnPython

@learnpython4cbse · 331 subscribers · 91 videos

We are introducing a channel in order to help children learn skills and also imbibe the sub ...more

learnpython4cbse.com and 2 more links

Customize channel | Manage videos

Home | Videos | Live | Playlists | Community

For You

Data file handling class 12 one shot | Data file Handling for class 12

One Shot Computer Networking with PYQs | cs class 12 | IP

Class 11 IP Revision Assignment - 1 | Class 11 Informatics Practices 2024

S
In one shot Video
Class 12: Comp. Sc.(083) CBSE BOARD
Class 12 COMP. SC
TOPIC: DATA FILE HANDLNG
With Solve Previous Years Board Questions
✓Text File
✓Binary File
✓CSV File
Total 15 Marks Cover Out of 40 Marks (from python)
Get full marks in Data file handling (15/15)
TIPS FOR ANNUAL EXAM FOR CLASS 12
1:11:22
MUST WATCH
Computer Networking
With Solve Previous Years Board Questions
In one shot Video Full Revision
Class 12 COMP. SC
Get full marks in Computer Networking (10/10)
WITH MCQ AND PRACTICE QUESTIONS
1:09:58
For Class 11 and 12 CBSE Board
Revision Assignment-1
Class 11 IP (2023-24)
Keynotes
✓How to get 70/70
✓Solution of Sample Paper -2
✓How to attempt Error finding questions and output question
✓Tips for writing Answers
TIPS FOR ANNUAL EXAM FOR CLASS 11
12:15

MARKING SCHEME - SAMPLE PAPER – 1

Code: Learnpython4cbse-1/5(25CBSE01)

INFORMATICS PRACTICES 12TH (CODE 065)

General Instructions:

Time: 3 Hrs. **Max. Marks: 70**

1. Please check this question paper contains 37 questions.
2. All questions are compulsory. However, internal choices have been provided in some questions. Attempt only one of the choices in such questions.
3. The paper is divided into 5 Sections- A, B, C, D and E.
4. Section A consists of 21 questions (1 to 21). Each question carries 1 Mark.
5. Section B consists of 7 questions (22 to 28). Each question carries 2 Marks.
6. Section C consists of 4 questions (29 to 32). Each question carries 3 Marks.
7. Section D consists of 2 case study type questions (33 to 34). Each question carries 4 Marks.
8. Section E consists of 3 questions (35 to 37). Each question carries 5 Marks.
9. All programming questions are to be answered using Python Language only.
10. In case of MCO, text of the correct answer should also be written.

Section A: Objective Questions (1 Mark Each)

1. **True or False:**
   - Statement: "A Pandas Series can only contain numeric data types."
   - **Answer**: False
   - Explanation: A Pandas Series can contain heterogeneous data types such as integers, floats, strings, etc.
2. **Counting characters in an expression:**
   - **Answer**: (a) LENGTH
   - Explanation: The LENGTH() function in SQL returns the number of characters in a string.
3. **Device amplifying signals:**
   - **Answer**: (c) Repeater
   - Explanation: Repeaters amplify signals to cover larger distances in networks.

4. **Display current date and time:**
   - **Answer**: (c) SYSDATE()
     - Explanation: In SQL, the SYSDATE() function returns the current system date and time.
5. **Digital footprint creation:**
   - **Answer**: (d) All of these
     - Explanation: Visiting websites, sending emails, and posting online create digital footprints.
6. **Concatenating DataFrames vertically:**
   - **Answer**: (c) pd.concat([dfl, df2])
     - Explanation: The pd.concat() function appends DataFrames either row-wise or column-wise.
7. **Incorrect statement regarding Data Visualization:**
   - **Answer**: (b) Visualizing large and complex data does not produce effective results.
     - Explanation: Data visualization simplifies complex data, making it interpretable.
8. **Using NULL in mathematical operations:**
   - **Answer**: False
     - Explanation: NULL values cannot participate in mathematical operations as they represent unknown data.
9. **CSV parameter for column headers in pd.read_csv():**
   - **Answer**: (a) header=True
     - Explanation: Setting header=True ensures the first row of the CSV file is treated as headers.
10. **Example of active digital footprint:**
    - **Answer**: (a) Sharing personal information on Facebook, Instagram, etc.
      - Explanation: Active digital footprints are created intentionally through online activities.
11. **SQL clause to filter records:**
    - **Answer**: (c) WHERE
      - Explanation: The WHERE clause filters records based on specified conditions.

12. **Collection of interlinked web pages:**
    - **Answer**: (c) World Wide Web
    - Explanation: The World Wide Web (WWW) is an interconnected network of web pages.
13. **Columns in DataFrame DI:**
    - **Answer**: (b) 2
    - Explanation: The DataFrame DI created has 2 columns corresponding to the given input.
14. **Correct netiquette examples:**
    - **Answer**: (d) All of these
    - Explanation: Respecting others' privacy, avoiding cyberbullying, and copyright adherence are netiquettes.
15. **Commands to display values of a series:**
    - **Answer**: (d) Both (a) and (b)
    - Explanation: school[0:] and school[:] both display all values in a Pandas Series.
16. **Matching SQL functions with descriptions:**
    - **Answer**: (c) P-2, Q-1, R-3, S-4
    - Explanation: LOWER(), LENGTH(), UPPER(), and TRIM() perform the respective described operations.
17. **Operator to combine SELECT query results:**
    - **Answer**: (b) UNION
    - Explanation: The UNION operator combines results of multiple SELECT queries.
18. **Horizontal box plot in Matplotlib:**
    - **Answer**: (c) plt.boxplot(data, vert=False)
    - Explanation: The vert=False parameter rotates the box plot horizontally.
19. **Characteristic of peer-to-peer networks:**
    - **Answer**: (b) Direct communication between devices
    - Explanation: Peer-to-peer networks do not rely on centralized servers.
20. **Assertion and Reason on Pandas head() function:**
    - **Answer**: (c) Assertion (A) is True, but Reason (R) is False

- Explanation: pandas.head() displays the first 5 rows, not the last 3.

21. **Assertion and Reason on SQL ALTER TABLE:**
    - **Answer**: (d) Assertion (A) is False, but Reason (R) is True
    - Explanation: The ALTER TABLE statement modifies the structure of a table, while DROP TABLE deletes it.

Section B: Short Answers (2 Marks Each)

Q22: Create a Pandas Series from a tuple:

```
import pandas as pd
data = (100, "Aavya Verma", 96.3, "A")
series = pd.Series(data)
print(series)
```

OR

Carefully observe the following code:

```
import pandas as pd
Year1 = {"L1": 5000, "L2": 8000, "L3": 2000, "L4": 18000}
Year2 = {"X": 13000, "Y": 14000, "Z": 12000}
totSales = {1: Year1, 2: Year2}
df = pd.DataFrame(totSales)

# Answers
i) print("Index:", df.index)
ii) print("Columns:", df.columns)
```

Q23:

- **Digital Footprints**: The trail of data users leave behind from online activities such as website visits and social media posts.
- **Phishing**: A cyber-attack where users are tricked into revealing confidential information like passwords or financial details.

Q24: Identify the SQL query error:

- **Error: The WHERE clause is misplaced; it should be before GROUP BY.**
- **Correct Query:**

```
SELECT HOUSE, COUNT(*)

FROM STUDENT

WHERE HOUSE='Green' OR HOUSE='Orange'

GROUP BY HOUSE;
```

Q25:

(A) Difference Between Web Browser and Web Server:

Feature	Web Browser	Web Server
Definition	A client application to access web pages	Software/hardware to store/serve web pages
Examples	Chrome, Firefox	Apache, IIS

(B) Role of Router in Networks:

- Directs data packets between networks.
- Ensures optimal routes for data transfer.

Q26: Four Date/Time Functions:

- **CURDATE()**: Returns the current date.
- **NOW()**: Returns the current date and time.
- **DATEDIFF(date1, date2)**: Returns days between two dates.
- **YEAR(date)**: Extracts the year from a date.

Q27: Intellectual Property Rights (IPR):

- Legal rights protecting creations of the mind (e.g., inventions, music, logos).

- Types: Copyrights, trademarks, patents.

Q28: Correct Python code:

- **Errors**: Key-value syntax, invalid slicing.

```
import pandas as pd
data = {'A': [1, 2, 3, 4, 5], 'B': [6, 7, 8, 9, 10]}
df = pd.DataFrame(data)
result = df.loc[1:4, ['A', 'B']]
print(result)
```

OR

(B) Complete the given Python code: to create a DataFrame from the given dictionary 'data' and display the first 3 rows using Pandas.

```
import pandas as pd

data = {'Name': ['John', 'Jane', 'Mike', 'Emily'],
'Age': [25, 30, 22, 28],
'City': ['New York', 'London', 'Paris', 'Sydney']}
df = pd.DataFrame(data)
print(df.head(3))
```

Section C: Analytical Questions (3 Marks Each)

Q29: Digital Footprint:

(i) Digital footprints are trails of user activity online. Simran's postings on social networks contribute to her online trace.

(ii) Risks: Identity theft, targeted advertising, cyberbullying.

(iii) Protection: Avoid oversharing, use privacy controls, delete unused accounts.

Q30: Create DataFrame from Dictionary:

```
 import pandas as pd
data = {'Name': {'A': 'Anamay', 'B': 'Aditi', 'C': 'Mehak', 'D': 'Kriti'},
'Class': {'A': 'XI', 'B': 'XI', 'C': 'XI', 'D': 'XI'},
'Marks': {'A': 95, 'B': 82, 'C': 65, 'D': 45}}
df = pd.DataFrame(data)
print(df)
```

OR

Write the Python Program to a Pandas Series as shown below using a list. :-

```
 import pandas as pd
data = {'Jammu': 44, 'Srinagar': 37, 'Delhi': 46, 'Agra': 48, 'Bangalore': 32}
series = pd.Series(data)
print(series)
```

Q31: SQL Table Creation and Data Insertion:

1. **SQL Statement to Create Table EMPLOYEES**:

```
CREATE TABLE EMPLOYEES (
EmployeeID INT PRIMARY KEY AUTO_INCREMENT,
FirstName VARCHAR(50),
LastName VARCHAR(50),
DateOfBirth DATE,
Salary DECIMAL(10, 2));
```

2. **SQL Query to Insert Data**:

```
INSERT INTO EMPLOYEES (EmployeeID, FirstName, LastName, DateOfBirth, Salary)
VALUES (1, 'Ronit', 'Rawat', '1985-09-25', 55000.00);
```

Q32: SQL Queries for Table Operations

(A) Given the table Drinks:

1. **Last 4 Characters in Uppercase**:

```
SELECT UPPER(SUBSTRING(DrinkName, -4)) AS Last4Chars FROM Drinks;
```

2. **Double the Cost**:

```
SELECT DrinkName, Cost * 2 AS DoubledCost FROM Drinks;
```

3. **Type and Cost of Drinks with "cola" in Name**:

```
SELECT Type, Cost FROM Drinks
WHERE DrinkName LIKE '%cola%';
```

OR

(B) Given the table Student:

1. **Students with Stream as Computer**:

```
SELECT Name FROM Student WHERE Stream = 'Computer';
```

2. **Count of Female Students**:

```
SELECT COUNT(*) AS FemaleCount FROM Student WHERE Sex = 'F';
```

3. **Stream-wise Count of Students**:

```
SELECT Stream, COUNT(*) AS StreamCount FROM Student GROUP BY Stream;
```

Section D: Case Studies (4 Marks Each)

Q33: Histogram Code for Math Test Scores

1. **Code to Generate Histogram**:

```
import matplotlib.pyplot as plt
# Data
scores = [55, 70, 85, 60, 90, 75, 80, 95, 65, 88]
# Histogram
plt.hist(scores, bins=5, color='blue', edgecolor='black')
# Titles and Labels
plt.title('Student Scores in Math Test')
plt.xlabel('Score Range')
```

```
    plt.ylabel('Frequency')
    # Save the Figure
    plt.savefig('names_scores_histogram.png')
    plt.show()
```

Q34: SQL Queries for Fitness Product Operations

(A) Table: FITNESS

1. **Products with Price > 20000**:

```
SELECT PName FROM FITNESS WHERE Price > 20000;
```

2. **Products by Manufacturer 'Aone'**:

```
SELECT PName FROM FITNESS WHERE Manufacturer = 'Aone';
```

3. **Reduce Prices by 25%**:

```
UPDATE FITNESS SET Price = Price * 0.75;
```

4. **Add New Product**:

```
INSERT INTO FITNESS (PCode, PName, Price, Manufacturer)
VALUES ('P7', 'Vibro Exerciser', 28000, 'Aone');
```

(B) Table: PRODUCT

1. **Average Price of Maharaja Products**:

```
SELECT AVG(Price) AS AvgPrice FROM PRODUCT WHERE BrandName = 'Maharaja';
```

2. **Count Distinct Brands**:

```
SELECT COUNT(DISTINCT BrandName) AS UniqueBrands FROM PRODUCT;
```

3. **Maximum Price**:

```
SELECT MAX(Price) AS MaxPrice FROM PRODUCT;
```

4. **Products with Quantity >= 10**:

```
SELECT ProductName FROM PRODUCT WHERE Quantity >= 10;
```

Section E: Complex Problem Solving (5 Marks Each)

Q35: Networking Setup and Justifications

1. **Network Types**:
   - (a) **Head Office and Tech Office**: LAN (distance: 70 m)

- (b) **Head Office and Coimbatore Office**: WAN (distance: 1952 m)

2. **Devices for Local Networks**:

- Suggest **Switch/Hub** for connecting computers within an office.

3. **Wiring Layout for New Delhi Offices**:

- Star topology: Central hub at Head Office connecting Sales and Tech Offices.

4. **Device for Data Security**:

- Use **Firewall Software** at the Head Office.

5. **Device and Protocol for Wireless Internet**:

- Device: **Router**
- Protocol: **Wi-Fi (802.11ac)**

Q36: DataFrame Operations

Given DataFrame SALES:

1. **Total Sales per City**:

```
print(SALES.sum(axis=1))
```

2. **City with Maximum Guava Sales**:

```
print(SALES['Guava'].idxmax())
```

3. **Total Sales for Each Fruit**:

```
print(SALES.sum(axis=0))
```

4. **City with Minimum Banana Sales**:

```
print(SALES['Banana'].idxmin())
```

5. **Average Sale for Each Fruit**:

```
print(SALES.mean(axis=0))
```

Q37: SQL Query Solutions

1. **Extract 7 Characters from Position 8**:

```
SELECT SUBSTRING('SHINING WORLD', 8, 7) AS ExtractedText;
```

2. **Position of 'Web' in String**:

```
SELECT INSTR('Dynamic Website', 'Web') AS Position;
```

3. **Round Value to 1 Decimal**:

```
SELECT ROUND(77.78, 1) AS RoundedValue;
```

4. **Remainder of 102 Divided by 9**:

```
SELECT MOD(102, 9) AS Remainder;
```

5. **Remove Leading and Trailing Spaces**:

```
SELECT TRIM(EMPID) AS TrimmedEMPID FROM EMP;
```

OR

Functions with Examples:

- **UCASE()**: Converts string to uppercase.

```
SELECT UCASE('hello') AS UppercaseText;
```

- **RTRIM()**: Removes trailing spaces.

```
SELECT RTRIM('hello ') AS TrimmedText;
```

- **MID()**: Extracts substring.

```
SELECT MID('hello', 2, 3) AS Substring;
```

- **INSTR()**: Finds position of a substring.

```
SELECT INSTR('Dynamic Website', 'Web') AS Position;
```

- **MOD()**: Finds remainder.

```
SELECT MOD(102, 9) AS Remainder;
```