

SAMPLE PAPER –2 MARKING SCHEME SUBJECT: COMPUTER SCIENCE

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 here¹](#).

Happy learning! .

Please like, Subscribe and share the Channel



SuperNova-LearnPython

@learnpython4cbse · 468 subscribers · 101 videos

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

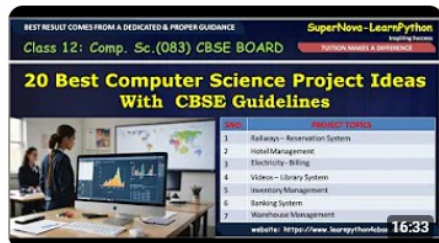
[learnpython4cbse.com](https://www.learnpython4cbse.com) and 2 more links

Customize channel

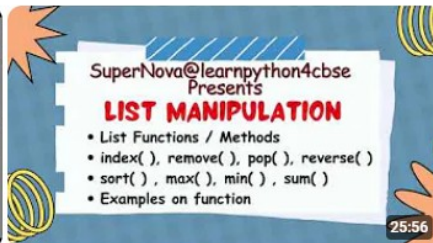
Manage videos

Home Videos Live Playlists Community

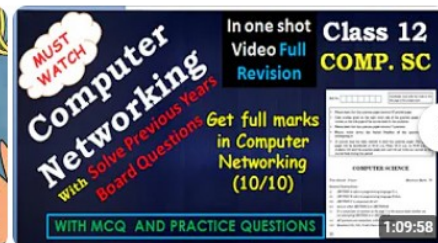
For You



20 BEST COMPUTER SCIENCE PROJECT IDEAS FOR CLASS 12 WITH GUIDELINES
244 views · 3 days ago



Class 11 CBSE | List Manipulation in Python | List Functions | List Methods | | Class11 IP
34 views · 11 months ago



One Shot Computer Networking with PYQs | cs class 12 | IP
489 views · 7 months ago

SAMPLE PAPER – MARKING SCHEME

Code: [Learnpython4cbse-2/5\(25CBSE02\)](#)

COMPUTER SCIENCE 12TH (CODE 083)

General Instructions:

Time: 3 Hrs.

Max. Marks: 70

1. 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 3 questions (29 to 31). Each question carries 3 Marks.
7. Section D consists of 4 questions (32 to 35). Each question carries 4 Marks.
8. Section E consists of 2 questions (36 to 37). Each question carries 5 Marks.
9. All programming questions are to be answered using Python Language only.
10. In case of MCQ, text of the correct answer should also be written.

SECTION A

[21x1= 21]

1	True	1
2	(a) ['xy','yz']	1
3	(a) False	1
4	(c) [9, 5, 6, 7, 8]	1
5	t	1
6	[3, 4]	1
7	0 1 2	1
8	(b) Removes the first occurrence of a specified value from the list	1
9	(c) Calc (Y=25)	1
10	pickle	1
11	True	1

12	(d) Error	1
13	DROP TABLE	1
14	(b) COUNT	1
15	(a) SQL statements can not be typed in lowercase or uppercase letter. SQL statements are case sensitive.	1
16	(b) Display records in increasing order of roll number	1
17	(c) Web hosting Or (b) T2[::-1]	1
18	(c)HTTP	1
19	Circuit switching	1
20	(a) Both A and R are true and R is the correct explanation for A.	1
21	(c) A is true and R is false.	1

SECTION B

[7x2= 14 Marks]

22	<pre>def reverseWordSentence(Sentence): words = Sentence.split("") newWords = [word[::-1] for word In words] newSentence = "".join(newWords) return newSentence Sentence = "Board exams are coming" pri nt(reverseWordSentence(Sentence))</pre>	2
23	(i) hex() (ii) range()	2
24	(I) (A) T1.count(10) OR (B) T1.index(25) (II) (A)T3 = T1 + T2 OR (B) T2[::-1]	2

25	Possible Outputs: (A) S-u-c- (B) S-u-c-c- (C) S-u-c-c-e- Minimum possible value of b: 2 Maximum possible value of b: 5	2
26	Constraints are check or conditions given on field of a table to restrict entry of invalid data example Unique, NOT NULL. The foreign key constant can be used to create a link between two tables by assigning one of the column as foreign key in a table.	2
27	(I) (A) The keyword used to define a view that combines data from multiple tables is CREATE VIEW <p style="text-align: center;">OR</p> (B) The clause used to filter records based on specific conditions is WHERE . (II) (A) ALTER TABLE Employees ADD Email VARCHAR(255); OR (B) ALTER TABLE Departments RENAME COLUMN OldName TO NewName;	2
28	(i) Wi-Fi is a wireless communication technology that allows devices to connect to the internet and exchange data without the need for physical cables. It uses radio waves to transmit information between devices, enabling wireless internet access and network connectivity for various devices like smartphones, laptops, and smart home devices. (ii) In guided transmission media, cables (or wires) are used for communicating data while in unguided transmission, in place of cables, signals travel through air. <p style="text-align: center;">OR</p> (i) Following are the three main types of computer networks, based upon the geographical area as follows: (a) Local Area Network (LAN) (b) Metropolitan Area Network (MAN) (c) Wide Area Network (WAN) (ii) A switch is a networking device, which is used to connect different LAN segments of a network. A switch also supports packet filtering and forwarding between LAN segments.	2

SECTION C

[3x3= 9 Marks]

29	<p>(A)</p> <pre>def write_urgent_lines(): try: with open('Messages.txt', 'r') as input_file: with open('UrgentLines.txt', 'w') as output_file: for line in input_file: if 'urgent' in line.lower(): output_file.write(line) print("Urgent lines have been written to 'UrgentLines.txt'.") except FileNotFoundError: print("The file 'Messages.txt' was not found.")</pre> <p style="text-align: center;">OR</p> <p>(B)</p> <pre>def count_lines_in_file(): try: with open('Paragraph.txt', 'r') as file: lines = file.readlines() print(f"Number of lines in 'Paragraph.txt': {len(lines)}") except FileNotFoundError: print("The file 'Paragraph.txt' was not found.")</pre>	3
30	<pre>def push(stack, item): stack.append(item) def pop(stack): if len(stack) == 0: return "Stack is empty" return stack.pop() def reverse_string(input_string): stack = [] # Push all characters of the string into the stack for char in input_string: push(stack, char) reversed_str = "" # Pop characters from the stack to reverse the string while len(stack) > 0: reversed_str += pop(stack) return reversed_str # Example usage input_string = "Hello" print("Reversed string:", reverse_string(input_string))</pre> <p style="text-align: center;">OR</p>	3

	<pre> (A) R={"OM":76, "JAI":45, "BOB":89, "AU": 65 , "ANU": 90, "TOM" :82} def PUSH(S,N): S.append(N) def POP(S): if S!=[]: return S.popO else: return None ST=[] for k in R: if R[k]>=75: PUSH(ST,k) while True: if ST!=[]: print(POP(ST),end="") else: break </pre>	
--	---	--

31	<p>(i) To list the BookTitle of FND type.</p> <pre>SELECT BookTitle FROM LIBRARY WHERE Type = 'FND';</pre> <p>(ii) To display a report listing BookTitle, Type and Price in descending order of price.</p> <pre>SELECT BookTitle, Type, Price FROM LIBRARY ORDER BY Price DESC;</pre> <p>(iii) To count the number of BookTitle, Who have FND type.</p> <pre>SELECT COUNT(BookTitle) FROM LIBRARY WHERE Type = 'FND';</pre> <p style="text-align: center;">OR</p> <p>(B) To insert a new row in the table LIBRARY.</p> <pre>INSERT INTO LIBRARY (No, BookTitle, Type, Publication, Quantity, Price) VALUES (7, 'Windows 8 Basics', 'FND', 'McGraw', 7, 150);</pre>	3
-----------	---	----------


```
import csv
def display_small_cities():
    with open('HealthSurvey.csv', 'r') as file:
        reader = csv.reader(file)
        # Skip the header if present
        next(reader)
        for row in reader:
            city = row[0]
            residents = int(row[1])
            if residents < 2000000:
                print(row)

# Example Usage
display_small_cities()
```

(II) Write a function to count the number of records in the file:

```
import csv
def count_records():
    with open('HealthSurvey.csv', 'r') as file:
        reader = csv.reader(file)
        # Skip the header if present
        next(reader)

    # Convert the reader object to a list and count the number of rows
    records = list(reader)
    count = len(records)

    print(f"Number of records: {count}")
```


	<p># Example Usage</p> <pre>count_records()</pre>	
34	<p>(a) SELECT NAME FROM STUDENTS WHERE ADDRESS IS NULL;</p> <p>(b) SELECT COUNT(PHONE) FROM STUDENTS;</p> <p>(c) SELECT GAME FROM SPORTS WHERE COACHNAME LIKE "%RAVI";</p> <p>(d) (A) SELECT * FROM SPORTS;</p> <p style="text-align: center;">OR</p> <p>(B) ALTER TABLE SPORTS ADD TOURNAMENT VARCHAR(20);</p>	4
35	<pre>def AddAndDelete(): import mysql.connector as mycon mydb=mycon.connect(host="localhost",user="root",passwd=" Pencil ", database=" ITEMDB ") mycur=mydb.cursor() no=int(input("Enter Student Admission Number: ")) nm=input("Enter Student Name: ") pr=float(input("Enter percentage: ")) cls=int(input("Enter class: ")) query="INSERT INTO student VALUES ({},{},{},{})";" query=query.format(no,nm,pr,cls) mycur.execute(query) mydb.commit() try: mycur.execute("Delete from student where ItemName='Pencil' ;") mycur.commit() except: mydb.close()</pre>	4

**SECTION E****[2x5= 10 Marks]****36** (a) tell() method tells you the current position of cursor within the file.

Syntax: file_object.tell()

(b) import pickle

def AddRegistration() :

f=open("Registration.dat","ab")

ans='y'

rno=" "

sname=" "

class=" "

Nos=0

while ans=='y':

rno=input("Enter registration no..")

sname=input("Enter student name ")

class=input("Enter class")

Nos=int(input("Enter number of subjects :"))

reglst=[rno,sname,class,Nos]

pickle.dump(reglst,f)

ans=input("Continue(y/n)")

f.close()

def CountRegistrationst):

f=open("Registration.dat","rb")

count=0

try:

while True:

phonelst=pickle.load(f)

count+=1

print("Number of registrations ",count)

except EOFError:

break

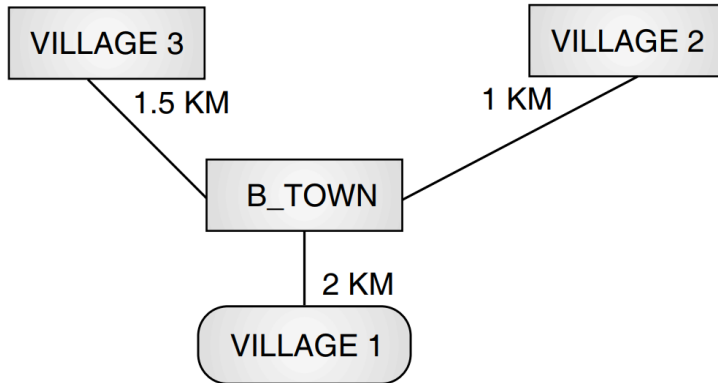
f.close()

AddRegistration()

CountRegistrations()

5

- 37** (i) B_TOWN can house the server as it has the maximum no. of computers.
(ii) Optical fibre cable is the best for this star topology.



- (iii) Switch
(iv) VoIP
(v) MAN (Metropolitan Area Network) will be formed as network would span within a city.

OR

- a) Video Conferencing

5