



**KENDRIYA VIDYALAYA SANGATHAN  
BHOPAL REGION**

**QUICK RIVISION MATERIAL**

त्वरित समीक्षा सामग्री



**Computer Science (083)**

**CLASS XII**

**2024-25**

केन्द्रीय विद्यालय सगठन

KENDRIYA VIDYALAYA SANGATHAN

**BHOPAL REGION**

**QUICK RIVISION MATERIAL 2024-25**

त्वरित समीक्षा सामग्री 2024-25

**CHIEF PATRON**



**Dr R SENTHIL KUMAR**  
**DEPUTY COMMISSIONER BHOPAL REGION**

**PATRON**



**SMT. KIRAN MISHRA**  
ASSISTANT COMMISSIONER  
KVS RO BHOPAL



**SMT. NIRMALA BUDANIA**  
ASSISTANT COMMISSIONER  
KVS RO BHOPAL



**SHRI VIJAY VEER SINGH**  
ASSISTANT COMMISSIONER  
KVS RO BHOPAL

**CO- ORDINATOR**  
**MRS PUNITA JYOTI**  
**PRINCIPAL KV ITBP SHIVPURI**

**PREPARED BY**  
**YASHDEV HADA**  
**PGT CS KV KHANDWA**

### 3. Distribution of Marks:

| Unit No. | Unit Name                                  | Marks |
|----------|--|-------|
| I        | Computational Thinking and Programming – 2 | 40    |
| II       | Computer Networks                          | 10    |
| III      | Database Management                        | 20    |
|          | Total                                      | 70    |

## UNIT – 1 Computational Thinking and Programming – 2

Python is a high-level, interpreted programming language known for its simplicity, readability, and versatility. It was created by Guido van Rossum and first released in 1991. Python emphasizes code readability and allows developers to express concepts in fewer lines of code compared to many other programming languages.

### Key Features of Python:

1. Simple and Readable Syntax
2. Interpreted Language
3. Dynamically Typed
4. Cross-Platform
5. Extensive Standard Library
6. Large Ecosystem of Third-Party Libraries
7. Versatile
8. Community Support

### Why Learn Python?

1. Beginner-Friendly
2. High Demand
3. Rapid Development
4. Open-Source.

### . Basics of Python Syntax

- Print Statements:  
`print("Hello, KVS Bhopal Region")`
- Comments: `# This is a comment`
- Indentation: Python uses indentation instead of braces for code blocks.

### 2. Variables and Data Types

- Variables: Dynamic typing, e.g.,  
`x = 10`  
`name = "Suyash".`
- Data Types:
  - Integers: 5
  - Floats: 5.0
  - Strings: "Hello"
  - Booleans: True, False

### 3. Basic Operators

- Arithmetic: +, -, \*, /, %, \*\* (power)
- Comparison: ==, !=, <, >, <=, >=
- Logical: and, or, not

### 4. Input and Output

- Input: `input("Enter your name: ")`
- Output: `print("Hello", name)`

### 5. Control Flow

- Conditional Statements:

**if condition:**

`# code block`

**elif another\_condition:**

`# another block`

**else:**

`# else block`

- Loops:

- For : Repetitive statement for finite number known to user(Example)

`for i in range(5):`

`print(i)`

- while: When no of iteration is not known to user

`while True:`

`print("hello")`

`ch = input("Want to`

`continue)`

### 6. Functions

- Define and call functions:

Example:

`def demo(name):`

`return f"Hello, {name}!"`

`print(demo("Ajay"))`

### 7. Data Structures

- Lists: `fruits = ["apple", "banana", "cherry"]`
- Tuples: `coordinates = (10, 20)`
- Dictionaries: `person = {"name": "Alina", "age": 18}`

8. File handling in Python allows you to work with files (read, write, append, etc.). It's straightforward and follows the principle of opening, working with, and closing files.

### File Handling Basics

1. Opening a File:  
Use the `open()` function to open a file. It requires a file name and a mode.

Example:

`file = open("filename.txt", "mode")`

Common Modes:

- 'r': Read (default mode). File must exist.
- 'w': Write. Creates a new file or overwrites an existing one.
- 'a': Append. Adds content to the end of a file.
- 'x': Create. Creates a file; raises an error if it already exists.

## 2. Reading Files

- `read()`: Reads the entire file content.
- `readline()`: Reads one line at a time.
- `readlines()`: Reads all lines and returns them as a list. Example

```
file = open("example.txt", "r")
content = file.read()
print(content)
file.close()

# Read file line by line
file = open("example.txt", "r")
for line in file:
    print(line.strip()) # Removes
    extra newline
file.close()
```

## 3. Writing to Files

To write to a file, open it in write (w) or append (a) mode. Writing will overwrite existing content in write mode.

Example:

```
file = open("example.txt", "w")
file.write("Hello, World!\n")
file.write("This is Python file
handling.")
file.close()
```

## 4. Appending to Files

Appending adds content to the end of the file without overwriting existing data.

Example:

```
file = open("example.txt", "a")
file.write("\nAppended text.")
file.close()
```

## 5. File Handling Methods

| Method                        | Description                                      |
|-------------------------------|--|
| <code>open()</code>           | Opens a file.                                    |
| <code>read(size)</code>       | Reads size bytes; if omitted, reads entire file. |
| <code>readline()</code>       | Reads a single line.                             |
| <code>readlines()</code>      | Reads all lines as a list.                       |
| <code>write(string)</code>    | Writes a string to the file.                     |
| <code>writelines(list)</code> | Writes a list of strings to the file.            |
| <code>close()</code>          | Closes the file.                                 |

## 6 Opening a Binary File

To open a binary file, use the `open()` function with `b` appended to the mode.

Modes for Binary Files:

- `rb`: Read a binary file.

- `wb`: Write to a binary file (overwrites existing data).
- `ab`: Append to a binary file.
- `rb+`: Read and write to a binary file.

## 7. Load() and dump() method:

The `dump()` and `load()` methods are used in Python's `pickle` module to serialize and deserialize Python objects into binary format. Serialization (or "pickling") is the process of converting a Python object into a byte stream, while deserialization (or "unpickling") is the reverse process.

### Importing the pickle Module

To use `dump()` and `load()`, you need to import the `pickle` module: Example

```
import pickle
```

#### 1. `pickle.dump()` :

The `dump()` method writes a Python object to a binary file.

Syntax:

```
pickle.dump(obj, file)
```

- `obj`: The Python object to serialize.
- `file`: The binary file object opened in write mode (`wb`).

Example:

```
import pickle
data = {"name": "Alina", "age": 17, "city":
"Khandwa"}
with open("data.pkl", "wb") as file:
    pickle.dump(data, file)
```

#### 2. `pickle.load()` :

The `load()` method reads a binary file and deserializes it into a Python object.

Syntax:

```
obj = pickle.load(file)
```

- `file`: The binary file object opened in read mode (`rb`).

```
import pickle
```

```
# Reading the object from a binary file
```

```
with open("data.pkl", "rb") as file:
```

```
    loaded_data = pickle.load(file)
```

```
print(loaded_data) # Output: {'name': 'Alina',
'age': 17, 'city': 'Khandwa'}
```

Example :

A binary file "student.dat" has structure [rollno, name, marks].

- Write a user defined function `insertRec()` to input data for a student and add to `student.dat`.
- Write a function `searchRno(r)` in Python which accepts the student's rollno as parameter and searches the record in the file "student.dat" and

shows the details of student i.e. rollno, name and marks (if found) otherwise shows the message as 'No record found'.

```
(i)
import pickle
def insertRec():
    f=open("student.dat","ab")
    rollno = int (input("Enter Roll Number : "))
    name=input("Enter Name :")
    marks = int(input("Enter Marks : "))
    rec = [rollno, name, marks ]
    pickle.dump( rec, f )
    f.close()

(ii)
def searchRNo(r):
    f=open("student.dat","rb")
    flag = False
    while True:
        rec=pickle.load(f)
        if rec[0] == r :
            print(rec['Rollno'])
            print(rec['Name'])
            print(rec['Marks'])
            flag == True
    f.close()
```

CSV (Comma-Separated Values) files are simple text files used to store tabular data, where each line represents a row and values are separated by commas. Python provides the built-in csv module to handle CSV files easily.

### 1. Importing the csv Module

```
import csv
```

### 2. Writing to a CSV File

You can use csv.writer to write data to a CSV file.

Example: Writing Rows to a CSV File

```
import csv
# Data to write
data = [
    ["Name", "Age", "City"],
    ["Ajay", 25, "khandwa"],
    ["Bobby", 30, "Mundi"],
    ["Deepa", 35, "Chhanera"]
]
# Writing data to a CSV file
with open("people.csv", "w", newline="")
as file:
    writer = csv.writer(file)
    writer.writerows(data)
print("Data written to people.csv")
```

### 3. Reading from a CSV File

You can use csv.reader to read data from a CSV file.

Example: Reading Rows from a CSV File

```
import csv
# Reading data from a CSV file
with open("people.csv", "r") as file:
    reader = csv.reader(file)
    for row in reader:
        print(row)
```

### 4. Handling Custom Delimiters

If your CSV file uses a different delimiter (e.g., semicolons or tabs), specify it using the delimiter parameter.

Example: Reading a Semicolon-Delimited CSV File

```
import csv
# Reading a CSV file with a semicolon
delimiter
with open("semicolon_delimited.csv", "r")
as file:
    reader = csv.reader(file, delimiter=";")
    for row in reader:
        print(row)
```

5. Example Program: Reading and Writing Here's a full example combining reading and writing:

```
import csv
# Write data to a CSV file
data = [
    ["Name", "Age", "City"],
    ["Ajay", 25, "khandwa"],
    ["Bobby", 30, "Mundi"],
    ["Deepa", 35, "Chhanera"]
]
with open("example.csv", "w", newline="")
as file:
    writer = csv.writer(file)
    writer.writerows(data)
# Read data from the same CSV file
with open("example.csv", "r") as file:
    reader = csv.reader(file)
    for row in reader:
        print(row)
```

### 6. Key Points:

- `newline=""`: Prevents extra blank lines when writing to CSV on Windows.
- `csv.reader`: Reads each row as a list.
- `csv.DictReader`: Reads each row as a dictionary.
- `csv.writer`: Writes rows as lists.
- `csv.DictWriter`: Writes rows as dictionaries.

### DATA STRUCTURE:

*Data Structure* means organization of data. A data structure has well defined *operations*, *behaviour* and *properties*.

A *stack* is a linear list, also known as LIFO list. *push* is the term coined for insertion of elements in a Stack list. and *Pop* is the term for deletion of

elements from top of the stack. In the Stack, if a user tries to remove an element from the empty Stack, it is called *underflow of stack*.

Simple Program :

```
stack = []
MAX_SIZE = 10
def push(item):
    if len(stack) >= MAX_SIZE:
        print("Stack Overflow! ")
    else:
        stack.append(item)
        print(f"{item} pushed onto the stack.")
def pop():
    if is_empty():
        print("Stack Underflow! ")
        return None
    else:
        return stack.pop()
def peek():
    if is_empty():
        print("The stack is empty.")
        return None
    else:
        return stack[-1]
def is_empty():
    return len(stack) == 0
def display():
    if is_empty():
        print("The stack is empty.")
    else:
        print("Stack contents:", stack)
if __name__ == "__main__":
    push(10)
    push(20)
    push(30)
    display()
```

PRACTICE QUESTIONS BASED ON PREVIOUS YEARS SESSION ENDING / COMPARTMENT EXAM

1.

Find and write the output of the following python code:

```
def fun(s):
    k=len(s)
    m=""
    for i in range(0,k):
        if(s[i].isupper()):
            m=m+s[i].lower()
        elif s[i].isalpha():
            m=m+s[i].upper()
        else:
            m=m+'bb'
    print(m)
fun('school2@com')
```

2. Rewrite the following code in python after removing all syntax error(s). Underline each correction done in the code:

```
a=5
work=true
b=hello
c=a+b
for i in range(10)
    if i%7=0:
        continue
```

3. Find and write the output of the following python code:

```
def display(s):
    l = len(s)
    m=""
    for i in range(0,l):
        if s[i].isupper():
            m=m+s[i].lower()
        elif s[i].isalpha():
            m=m+s[i].upper()
        elif s[i].isdigit():
            m=m+"$"
        else:
            m=m+"*"
    print(m) display("EXAM20@cbse.com")
```

4. Kavita wants to understand the concept of methods in a dictionary. Help her to find the answers of the following operations on a python dictionary:

```
d = {'M':10, 'N':20, 'O':30, 'P':40}
r = d.popitem()
print(r)
x = d.pop('N')
print(x)
print(list(d.keys()))
d.setdefault('X',60)
print(d)
```

5.

Find and write the output of the following Python code :

```
Data = ["P", 20, "R", 10, "S", 30]
Times = 0
Alpha = ""
Add = 0
for C in range(1, 6, 2):
    Times = Times + C
    Alpha = Alpha + Data[C-1]+"$"
    Add = Add + Data[C]
print Times, Add, Alpha
```

6. Write a python program that accepts a list of integers from user and prints all the integers that have 3 as the last digit.

Eg: for the list [10, 23, 3, 83, 93] the program should print 23 3 93

7. Write a program to read text file story.txt and count number of lines starting with letter 'A' or 'a'.

Solution:

```
F=open("story.txt",'r')
count=0
L=F.readlines()
for i in L:
    if i[0]=='A' or i[0]=='a':
        count=count+1
    print("no. of lines starting with a=",count)
F.close()
```

8. Write a program to read the file data.txt and count number of uppercase, lowercase in it.

```
F=open("data.txt",'r')
u=0
l=0
s=F.read()
for i in s:
    if i.isupper():
        u=u+1
    if i.islower():
        l=l+1
print("Number of uppercase characters=",u)
print("Number of lowercase characters=",l)
F.close()
```

9. Write a program to read the file data.txt and count number of spaces in it.

```
F=open("data.txt",'r')
space=0
s=F.read()
for i in s:
    if i.isspace():
        space=space+1
print("Number of spaces=",space)
F.close()
```

10. Write a program to read the file article.txt and count occurrences of words "the" in the file.

```
F=open("article.txt",'r')
count=0
s=F.read()
L=s.split()
for word in L:
    if word=="the":
        count=count+1
print("No. of occurrences of word the=",count)
F.close()
```

11. Write a method/function countln() in python to read lines from a text file rept.txt, and count those lines which are starting either with 'E' and starting with 'T' respectively. Display the Total count separately.

```
def countln():
    f=open("rept.txt")
    d=f.readlines()
```

```
le=0
lt=0
for i in d:
    if i[0]=='E':
        le=le+1
    elif i[0]=='T':
        lt=lt+1
print("no of line start with",le)
print("no of line start with",lt)
```

12. Program to search the record of a particular student from CSV file on the basis of inputted name.

```
import csv
#input Roll number you want to search
number = input('Enter number to find: ')
found=0
#read csv, and split on "," the line
with open('Student_Details.csv') as f:
    csv_file = csv.reader(f, delimiter=",")
    #loop through csv list
    for row in csv_file:
        #if current rows index value (here 0) is equal
        #to input, print that row
        if number ==row[0]:
            print (row)
            found=1
        else:
            found=0
```

```
if found==1:
    pass
else:
    print("Record Not found")
```

13. Write a program to perform read and write operation onto a student.csv file having fields as roll number, name, stream and percentage.

```
import csv
with open('Student_Details.csv','w',newline='')
as csvf:
```

```
writcsv=csv.writer(csvf,delimiter=',')
choice='y'
while choice.lower()=='y':
    r1=int(input("Enter Roll No.: "))
    n=input("Enter Name: ")
    p=float(input("Enter Percentage: "))
    r=input("Enter Remarks: ")
    writcsv.writerow([r1,n,p,r])
    print(" Data saved in Student Details
    file..")
    choice=input("Want add more
    record(y/n)....")
```

```
with open('Student_Details.csv','r',newline='') as
fileobject:
```

```
readcsv=csv.reader(fileobject)
for i in readcsv:
    print(i)
```

14. Write a python program to implement a stack using a list data-structure.

```
def isempty(stk):
    if stk==[]:
```

```

    return True
else:
    return False
def push(stk,item):
    stk.append(item)
    top=len(stk)-1
def pop(stk):
    if isempty(stk):
        return "underflow"
    else:
        item=stk.pop()
    if len(stk)==0:
        top=None
    else:
        top=len(stk)-1
    return item
def peek(stk):
    if isempty(stk):
        return "underflow"
    else:
        top=len(stk)-1
        return stk[top]
def display(stk):
    if isempty(stk):
        print('stack is empty')
    else:
        top=len(stk)-1
        print(stk[top], '<-top')
        for i in range(top-1,-1,-1):
            print(stk[i])
#Driver Code
def main():
    stk=[]
    top=None
    while True:
        print("'stack operation
        1.push
        2.pop
        3.peek
        4.display
        5.exit'")
        choice=int(input('enter choice:'))
        if choice==1:
            item=int(input('enter item:'))
            push(stk,item)
        elif choice==2:
            item=pop(stk)
            if item=="underflow":
                print('stack is underflow')
            else:
                print('popped')
        elif choice==3:
            item=peek(stk)
            if item=="underflow":
                print('stack is underflow')
            else:
                print('top most item is:',item)
        elif choice==4:
            display(stk)
        elif choice==5:

```

```

        break
    else:
        print('invalid')
        exit()
main ()

```

## UNIT – 2 COMPUTER NETWORKS

### 1. What is a Computer Network?

A computer network is a group of interconnected devices (computers, servers, printers, etc.) that share data and resources. The connections can be wired (e.g., Ethernet cables) or wireless (e.g., Wi-Fi).

Advantages of Networks:

- **Resource Sharing:** Printers, files, and software can be shared.
- **Communication:** Email, instant messaging, video conferencing.
- **Data Sharing:** Users can share files and databases.
- **Cost-Effective:** Centralized management and resource sharing reduce costs.

### 2. Types of Networks

#### a) Based on Size and Geographical Area

1. **PAN (Personal Area Network):**
  - Smallest network (e.g., Bluetooth, USB).
  - Used for personal devices like laptops, smartphones, and wearables.
2. **LAN (Local Area Network):**
  - Covers a small geographical area (e.g., office, school).
  - High-speed and inexpensive.
3. **MAN (Metropolitan Area Network):**
  - Covers a city or a large campus.
  - Example: Cable TV network.
4. **WAN (Wide Area Network):**
  - Covers a large geographical area (e.g., the internet).
  - Connects multiple LANs and MANs.

#### b) Based on Communication Model

1. **Client-Server:**
  - Centralized server provides resources to clients.
  - Example: Email server, web server.
2. **Peer-to-Peer (P2P):**
  - All devices act as both clients and servers.
  - Example: File sharing between computers.

### 3. Components of a Network

#### 1. Hardware Components:



- Node: Any device connected to the network (e.g., computer, printer).
- Router: Connects multiple networks and routes data packets.
- Switch: Connects devices within a LAN.
- Modem: Converts digital data to analog for internet connection.
- NIC (Network Interface Card): Hardware for connecting to a network.

## 2. Software Components:

- Protocols: Rules for data communication (e.g., TCP/IP, HTTP).
- Operating System: Manages network resources (e.g., Windows Server, Linux).

---

## 4. Network Topologies

The physical or logical arrangement of network devices.

### 1. Bus Topology:

- All devices are connected to a single communication line.
- Pros: Simple, low cost.
- Cons: Failure in the bus affects the entire network.

### 2. Star Topology:

- All devices are connected to a central hub or switch.
- Pros: Easy to manage, reliable.
- Cons: Hub failure affects the network.

### 3. Ring Topology:

- Devices are connected in a circular manner.
- Pros: Simple, orderly communication.
- Cons: Failure in one device can disrupt the network.

---

## 5. Network Protocols

Protocols define rules for data communication.

### 1. TCP/IP (Transmission Control Protocol/Internet Protocol):

- Foundation of the internet.
- Ensures reliable data transmission.

### 2. HTTP/HTTPS (HyperText Transfer Protocol):

- Protocol for accessing web pages.
- HTTPS adds encryption for secure communication.

### 3. FTP (File Transfer Protocol):

- Used for transferring files between computers.

### 4. SMTP (Simple Mail Transfer Protocol):

- Used for sending emails.

### 5. DNS (Domain Name System):

- Converts domain names (e.g., www.google.com) into IP addresses.

---

## 6. Common Networking Devices

### 1. Hub:

- Broadcasts data to all devices.
- No intelligence.

### 2. Switch:

- Sends data to the intended recipient device.
- Smarter than a hub.

### 3. Router:

- Connects different networks.
- Used for internet access.

### 4. Gateway:

- Translates data between different network types.

### 5. Access Point:

- Provides wireless connectivity.

---

## 7. IP Addressing

### 1. IPv4:

- 32-bit address (e.g., 192.168.1.1).
- Limited to 4.3 billion addresses.

### 2. IPv6:

- 128-bit address (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334).
- Vastly larger address space.

---

## PREVIOUS YEARS BAORD QUESTION

1. Write two points of difference between Bus topology and star topology.

2. Write two points of difference between XML and HTML.

3. Write the full forms of the following:

(i) HTTP (ii) FTP (iii) SMTP (iv) CDMA

4. Discuss the use of REMOTE LOGIN

5. Write two advantages and two disadvantages of circuit switching.

6. Differentiate between Web server and web browser. Write any two popular web browsers.

7. Classify each of the following Web Scripting as Client Side Scripting and Server Side Scripting :

(i) Java Scripting

(ii) ASP

(iii) VB Scripting

(iv) JSP

8. What is Bandwidth? What is the measuring unit of Bandwidth in term of range of frequencies a channel can pass?

9 Write two differences between Coaxial and Fiber transmission media.

**IMPORTANT FULL FORM BASED ON PREVIOUS YEAR BOARD QUESTIONS**

|  |   |
|--|---|
| <b>ARPANET - Advanced Research Projects Agency Network</b> |   |
| <b>CDMA</b>  | - Code Division Multiple Access               |
| <b>FTP</b>   | - FILE TRANSFER PROTOCOL                      |
| <b>FSF</b>   | - FREE SOFTWARE FOUNDATION                    |
| <b>GPRS</b>  | - General Packet Radio Service                |
| <b>GNU</b>   | - GNU"s not Unix                              |
| <b>GSM</b>   | - Global System for Mobile (communication)    |
| <b>HTML</b>  | -Hyper Text Markup Language                   |
| <b>HTTP</b>  | - Hyper Text Transfer Protocol                |
| <b>MAN</b>   | -Metropolitan Area Network                    |
| <b>MODEM</b>   | - Modulator - Demodulator                     |
| <b>PPP</b>   | - Point To Point Protocol                     |
| <b>SMS</b>   | - Short Message/Messaging Service             |
| <b>SMTP</b>  | Simple Mail Transfer Protocol                 |
| <b>TCP/IP</b>  | - Transfer Control Protocol/Internet Protocol |
| <b>URL</b>   | - Uniform Resource Locator                    |
| <b>VoIP</b>  | - Voice Over Internet Protocol                |
| <b>WAN</b>   | - Wide Area Network                           |
| <b>WLL(WiLL)</b>   | - Wireless in Local Loop                      |
| <b>WWW</b>   | - World Wide Web                              |
| <b>XML</b>   | - eXtensible Markup Language                  |

10. Differentiate between XML and HTML.

| S.No. | HTML  | XHTML   |
|-------|---|---|
| 1.    | HTML stands for Hypertext Markup Language.  | XHTML stands for Extensible Hypertext Markup Language.                                |
| 2.    | It was developed by Tim Berners-Lee.  | It was developed by W3C i.e World Wide Web Consortium.                                |
| 3.    | It was developed in 1991.   | It was released in 2000.  |
| 4.    | It is extended from SGML.   | It is extended from XML and HTML.   |
| 5.    | The format is a document file format.   | The format is a markup language.  |
| 6.    | All tags and attributes are not necessarily to be in lower or upper case.           | In this, every tag and attribute should be in lower case.                             |
| 7.    | Doctype is not necessary to write at the top.                                       | Doctype is very necessary to write at the top of the file.                            |
| 8.    | It is not necessary to close the tags in the order they are opened.                 | It is necessary to close the tags in the order they are opened.                       |
| 9.    | While using the attributes it is not necessary to mention quotes. For e.g. <Geeks>. | While using the attributes it is mandatory to mention quotes. For e.g. <Geeks="GFG">. |
| 10.   | Filename extension used are .html, .htm.  | Filename extension are .xhtml, .xht, .xml.  |

11) What is the purpose of using a repeater in the context of networking?

12) What are repeaters?

A) A repeater is a network device that amplifies and restores signals for long distance transmission.

It is used in long network lines, which exceed the maximum rated distance for a single run.

Repeaters are of two types:

(i) Amplifier : amplifies all incoming signals over the network. (it amplifies both the signal and any concurrent noise)

(ii) Repeater : collected inbound packet and then retransmits the packet as if it were starting from the source station.

13) What is a Hub?

Ans) A Hub is used for a central connection between two or more computers on a network.

OR

A Hub is a network device used to connect two or more computers.

OR

A Hub is an unintelligent network device to connect computers.

Hubs are of two types:

(i) Active hubs: electrically amplify the signal as it moves from one connected device to another.

(ii) Passive hubs: allow the signal to pass from one computer to another without any change.

14) What is a Modem?

Or What is the purpose of using a MODEM?

Ans) Modem is a Modulation Demodulation device that converts analog signal to digital signal and vice versa.

15) What is a bridge?

A) A bridge is a device that lets you link two networks together. Bridges are smart enough to know which computers are on which side of the bridge, so they only allow those messages that need to get to the other side to cross the bridge. This improves performance on both sides of the bridge.

As a packet arrives at the bridge, the bridge examines the physical destination address of the packet. The bridge then decides whether or not to let the packet cross

16) CASE STUDY BASED QUESTION:

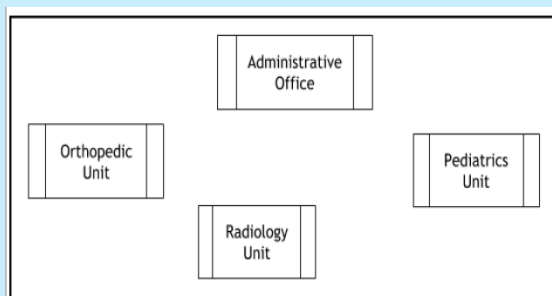
Ayurveda Training Educational Institute is setting up its centre in Hyderabad with four specialized departments for Orthopedics, Neurology and Pediatrics along with an administrative office in separate buildings. The physical distances between these department buildings and the number of computers to be installed in these departments and administrative office are given as follows.

You, as a network expert, have to answer the queries as raised by them in (i) to (iv)  
Shortest distances between various locations in meters:

|   |     |
|---|-----|
| Administrative office to Orthopedics Unit | 55  |
| Neurology Unit to Administrative Office   | 30  |
| Orthopedics Unit to Neurology Unit        | 70  |
| Pediatrics Unit to Neurology Unit         | 50  |
| Pediatrics Unit to Administrative Office  | 40  |
| Pediatrics Unit to Orthopedics Unit       | 110 |

Number of Computers installed at various locations are as follows:

|                       |     |
|-----------------------|-----|
| Pediatrics Unit       | 40  |
| Administrative Office | 140 |
| Neurology             | 50  |
| Orthopedics Unit      | 80  |

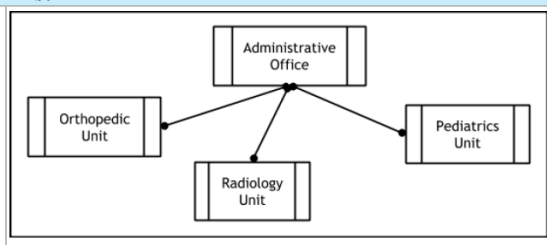


1. Suggest the most suitable location to install the main server of this institution to get efficient connectivity.

Ans: Administrative Office

2. Suggest the best cable layout for effective network connectivity of the building having server with all the other buildings.

Ans:



OR

Administrative Office is connected to Orthopedic, Radiology, Pediatrics units directly in a Star Topology

3. Suggest the devices to be installed in each of these buildings for connecting computers installed within the building out of the following:

\* Gateway \* Modem \* Switch

Ans: Switch

4. Suggest the topology of the network and network cable for efficiently connecting each computer installed in each of the buildings out of the following: Topologies: Bus Topology, Star Topology  
Network Cable: Single Pair Telephone, Coaxial Cable, Ethernet Cable.  
Topology : Star Topology  
Network Cable: Ethernet Cable / Coaxial Cable

### 17) CASE STUDY BASED QUESTION:

DEV PUBLIC SCHOOL in Darjeeling is setting up the network between its different wings. There are 4 wings named as SENIOR(S), JUNIORS (J), ADMIN (A) and HOSTEL (H). (2006OD)  
Distance between various wings is given below:  
Number of Computers

|        |     |
|--------|-----|
| Wing A | 10  |
| Wing S | 200 |
| Wing J | 100 |
| Wing H | 50  |

i) Suggest a suitable Topology for networking the computer of all wings

Ans) Star Topology OR Bus Topology

ii) Name the wing where the server to be installed. Justify your answer. 1m

Ans) Wing S as it has the maximum number of computers

OR Wing A as it is placed in the Admin Wing (for security reasons)

iii) Suggest the placement of Hub/Switch in the network.

Ans) Inside all the four wings

iv) Mention in economic technology to provide internet accessibility to all wings. 1m

Ans: Any one of the following:

Dialup, TCP/IP, DSL, Modem, Broadband, Cable, ISDN, Telephone Line, Co-axial, Ethernet Cable, Radiowave

## UNIT – 3 DATABASE MANAGEMENT

### 1. Introduction to DBMS

- **Definition:** A Database Management System is software that allows users to store, manage, and retrieve data efficiently.
- **Examples:** MySQL, PostgreSQL, Oracle DB, MongoDB.
- **Advantages:**
  - Data organization and integrity.
  - Reduces redundancy.

- Facilitates data sharing and security.

## 2. Database Concepts

- **Entity:** A real-world object (e.g., student, book).
- **Attribute:** A property or characteristic of an entity (e.g., Student\_Name, Roll\_No).
- **Primary Key:** A unique identifier for a record in a table.
- **Foreign Key:** A field in one table that refers to the primary key in another table.
- **Tuple:** A single row in a table.
- **Schema:** The structure of a database.

## 3. Relational Model

- Data is organized in tables (relations).
- Tables have rows (tuples) and columns (attributes).
- **Relational operations:** SELECT, INSERT, UPDATE, DELETE.

## 4. Structured Query Language (SQL)

SQL is used to interact with databases. Common SQL commands include:

- **DDL (Data Definition Language):** CREATE, ALTER, DROP.
- **DML (Data Manipulation Language):** INSERT, UPDATE, DELETE.
- **DQL (Data Query Language):** SELECT.
- **TCL (Transaction Control Language):** COMMIT, ROLLBACK.

### Common DDL Commands

1. **CREATE:** Used to create new database objects.

```
CREATE TABLE Students (
  RollNo INT PRIMARY KEY,
  Name VARCHAR(50),
  Age INT,
  Marks FLOAT
);
```

2. **ALTER:** Used to modify the structure of an existing table.

- Add a column:

```
ALTER TABLE Students ADD Email
VARCHAR(100);
```

- Modify a column:

```
ALTER TABLE Students MODIFY Age INT
NOT NULL;
```

- Drop a column:

```
ALTER TABLE Students DROP COLUMN
Email;
```

3. **DROP:** Used to delete a table or database.

- Drop a table:

```
DROP TABLE Students;
```

- Drop a database:

```
DROP DATABASE SchoolDB;
```

### Common SQL Queries:

- **Create Table:**

```
CREATE TABLE Students (
  RollNo INT PRIMARY KEY,
  Name VARCHAR(50),
  Class INT,
  Marks FLOAT
);
```

- **Insert Data:**

```
INSERT INTO Students (RollNo, Name, Class,
Marks)
```

```
VALUES (1, 'John Doe', 12, 85.5);
```

- **Select Data:**

```
SELECT * FROM Students WHERE Marks >
80;
```

- **Update Data:**

```
UPDATE Students SET Marks = 90 WHERE
RollNo = 1;
```

- **Delete Data:**

```
DELETE FROM Students WHERE RollNo = 1;
```

### VARIOUS CLAUSES IN SQL

SQL clauses are used to filter, sort, and manipulate data efficiently in queries. Here's a detailed explanation with examples for each:

#### 1. LIKE Clause

The LIKE clause is used in a WHERE statement to search for patterns in text data using wildcards.

##### Wildcards in LIKE:

- **%:** Represents zero or more characters.
- **\_:** Represents a single character.

##### Examples:

1. Find students whose names start with 'J':

```
SELECT * FROM Students WHERE Name
LIKE 'J%';
```

2. Find students whose names end with 'e':

```
SELECT * FROM Students WHERE Name
LIKE '%e';
```

3. Find students whose names contain 'an':

```
SELECT * FROM Students WHERE Name
LIKE '%an%';
```

4. Find students whose names have exactly 4 characters:

```
SELECT * FROM Students WHERE Name
LIKE '____';
```

#### 2. BETWEEN Clause

The BETWEEN clause is used to filter rows based on a range of values. It works for numeric, date, and text types.

##### Examples:

1. Find students with marks between 50 and 80:

```
SELECT * FROM Students WHERE Marks
BETWEEN 50 AND 80;
```

- Find records of students born between two dates:

```
SELECT * FROM Students WHERE DOB BETWEEN '2005-01-01' AND '2010-12-31';
```

### 3. DISTINCT Clause

The DISTINCT clause is used to retrieve unique values from a column, eliminating duplicates.

Examples:

- Find unique classes:

```
SELECT DISTINCT Class FROM Students;
```

- Find unique combinations of Class and Section:

```
SELECT DISTINCT Class, Section FROM Students;
```

### 4. ORDER BY Clause

The ORDER BY clause is used to sort the result set in ascending (ASC, default) or descending (DESC) order.

Examples:

- Sort students by marks in ascending order:

```
SELECT * FROM Students ORDER BY Marks;
```

- Sort students by marks in descending order:

```
SELECT * FROM Students ORDER BY Marks DESC;
```

- Sort by multiple columns (e.g., Class first, then Marks):

```
SELECT * FROM Students ORDER BY Class ASC, Marks DESC;
```

### 5. Other Common Clauses

#### GROUP BY Clause

Used to group rows with the same values in specified columns and perform aggregate functions.

```
SELECT Class, AVG(Marks) AS AverageMarks FROM Students GROUP BY Class;
HAVING Clause
```

Used to filter grouped records after GROUP BY.

```
SELECT Class, AVG(Marks) AS AverageMarks FROM Students GROUP BY Class HAVING AVG(Marks) > 70;
```

#### IN Clause

Used to filter rows where a column matches any value in a list.

```
SELECT * FROM Students WHERE Class IN (10, 12);
```

## PREVIOUS YEAR BOARD QUESTIONS

- What do you understand by Primary Key? Give a suitable example of Primary Key from a table containing some meaningful data.

Ans. An attribute or set of attributes which are used to identify a tuple uniquely is known as Primary Key.

Table: Item

| Ino | Item   | Qty |
|-----|--------|-----|
| I01 | Pen    | 300 |
| I02 | Pencil | 780 |
| I04 | CD     | 450 |
| I09 | Floppy | 700 |

PRIMARY KEY

- What is the importance of a primary key in a table? Explain with suitable example.

Ans: Primary Key: A primary key is a set of one or more attributes that can uniquely identify tuples within the relations.

A primary key comprises a single column or set of columns. No two distinct rows in a table can have the same value (or combination of values) in those columns. Depending on its designing, a table may have arbitrarily many candidate keys but at most one primary key. The primary key is non redundant. Ie it does not have duplicate values in the same relation.

Ex: Consider a table consists the following attributes:

AdmnNo, FirstName, LastName, SirName, M1, M2, M3, Total, Avg, FName

Here we can uniquely identify the rows in the relation with following key combinations:

- AdmnNo
- FirstName, LastName, SirName
- FirstName, LastName, FName, etc.

We can set any one of the above candidate keys as primary key, others are called as alternate keys.

- Give a suitable example of a table with sample data and illustrate Primary and Candidate Keys in it.

Ans A table may have more than one such attribute/group of attribute that identifies a row/tuple uniquely, all such attribute(s) are known as Candidate Keys. Out of the Candidate keys, one is selected as Primary Key.

Ex: Table: Stock

| Ino | Item   | Qty |
|-----|--------|-----|
| 101 | Pen    | 560 |
| 102 | Pencil | 780 |
| 104 | CD     | 450 |
| 109 | Floppy | 700 |
| 105 | Eraser | 300 |
| 103 | Duster | 200 |

Here: Ino – Primary Key  
Ino, Item – Candidate Keys,

4) Give a suitable example of a table with sample data and illustrate Primary and Alternate Keys in it.

Ans A table may have more than one such attribute/group of attribute that Identifies a row/tuple uniquely, all such attribute(s) are known as Candidate

Keys. Out of the Candidate keys, one is selected as Primary Key. while the rest are the Alternate Keys.

Ex: Table: Stock  
 Ino Item Qty  
 101 Pen 560  
 102 Pencil 780  
 104 CD 450  
 109 Floppy 700  
 105 Eraser 300  
 103 Duster 200

Here: Ino, Item – Candidate Keys, Ino – Primary Key

Item – Alternate Key

Explain the concept of candidate key with the help of an appropriate example.

5) What is the purpose of a key in a table? Give an example of a key in a table.

Ans) An attribute/group of attributes in a table that identifies each tuple uniquely is known as a Key.

OR

Any correct definition of Key / Primary Key / Candidate Key / Alternate Key

6) Differentiate between Candidate key and Primary key in context of RDBMS. )

Differentiate between Candidate Key and alternate Key in context of RDBMS.

Differentiate between primary key and alternate key.

What is an alternate key?

What do you understand by the terms primary key and degree of a relation in relational data base?

What do you understand by the candidate key and cardinality of a relation in relational data base?

What is primary key in a table?

Ans) Candidate Key: All attribute combinations inside a relation that can serve primary key are Candidate Keys as they are candidates for the primary key position.

Primary Key: A primary key is a set of one or more attributes that can uniquely identify tuples within the relations.

Alternate Key: A candidate key that is not the primary key is called an Alternate Key.

1. Write SQL queries for (i) to (xii) and find outputs for SQL queries (a) to (g), which are based on the tables.

TRAINER

| TID | TNAME      | CITY       | HIREDATE   | SALARY |
|-----|------------|------------|------------|--------|
| 101 | SUNAINA    | MUMBAI     | 1998-10-15 | 90000  |
| 102 | ANAMIKA    | DELHI      | 1994-12-24 | 80000  |
| 103 | DEEPTI     | CHANDIGARG | 2001-12-21 | 82000  |
| 104 | MEENAKSHI  | DELHI      | 2002-12-25 | 78000  |
| 105 | RICHA      | MUMBAI     | 1996-01-12 | 95000  |
| 106 | MANIPRABHA | CHENNAI    | 2001-12-12 | 69000  |

COURSE

| CID  | CNAME   | FEES  | STARTDATE  | TID |
|------|---------|-------|------------|-----|
| C201 | AGDCA   | 12000 | 2018-07-02 | 101 |
| C202 | ADCA    | 15000 | 2018-07-15 | 103 |
| C203 | DCA     | 10000 | 2018-10-01 | 102 |
| C204 | DDTP    | 9000  | 2018-09-15 | 104 |
| C205 | DHN     | 20000 | 2018-08-01 | 101 |
| C206 | O LEVEL | 18000 | 2018-07-25 | 105 |

WRITE SQL QUERIES

i) To Display the Trainer name, City from table Trainer.

Ans) SELECT TNAME, CITY from TRAINER;

ii) Display all details of table COURSE

Ans) SELECT \* FROM COURSE;

iii) Display the Trainer Name, City & Salary in descending order of their Hiredate.

Ans:

SELECT TNAME, CITY, SALARY FROM TRAINER ORDER BY HIREDATE DESC;

iv) To display all the details of those trainers whose name ends with „A“

Ans) select \* from Trainer where Tname like “%A”

v) Display all details from the table COURSE in ascending order of their STARTDATE

Ans)

SELECT \* FROM COURSE ORDER BY STARTDATE ASC;

vi) To display CName and Fees of those Courses Whose Fees range in between 10000-15000.

Ans) select CName, Fees from Course where Fees between 10000 and 15000;

vii) To display total salary of trainers from city name ends with “I”.

Ans) SELECT SUM(SALARY) FROM TRAINER WHERE CNAME LIKE “%I”;

viii) To display all Trainer Details from city “Mumbai”

Ans: SELECT \* FROM TRAINER WHERE CITY = ' Mumbai ' ;

ix) To display CName, Fees and StartDate of all Courses which are started before 15th July 2018

Ans) SELECT CNAME,FEES,STARTDATE FROM COURSE WHERE STARTDATE < '2018-07-15';

x) To display the last date (recent most) HIREDATE from the table TRAINER

Ans: SELECT MAX(HIREDATE) FROM TRAINER;

xi) To display the TNAME and CITY of Trainer who joined the Institute in the month of December 2001.

Ans) SELECT TNAME, CITY FROM TRAINER WHERE HIREDATE BETWEEN "2001-12-01" AND "2001-12-31";

OR

SELECT TNAME, CITY FROM TRAINER WHERE HIREDATE >= "2001-12-01" AND HIREDATE <= "2001-12-31";

OR

SELECT TNAME, CITY FROM TRAINER WHERE HIREDATE LIKE "2001-12%";

xii) To display TNAME, HIREDATE, CNAME, STARTDATE from tables TRAINER and COURSE of all those courses whose FEES is less than or equal to 10000.

Ans) SELECT TNAME,HIREDATE,CNAME,

(h) select TNAME, CITY, SALARY from TRAINER T, COURSE C where T.TID!=C.TID;

A) TNAME CITY SALARY

ManiPrabha Chennai

69000STARTDATE FROM TRAINER, COURSE WHERE TRAINER.TID=COURSE.TID AND FEES<=10000;

xiii) To display CNAME, Fees, TName,City from tbales Trainer and Course of all those persons whose Hiredate Before Year 2000.

Ans) SELECT CNAME, FEES, TNAME, CITY from TRAINER, COURSE WHERE TRAINER.TID=COURSE.TID AND HIREDATE<"2000-01-01";

(XiV) To display number of Trainers from each city.

Ans) SELECT CITY,COUNT(\*) FROM TRAINER;

### WRITE SQL OUTPUTS

(a) SELECT TID, TNAME, FROM TRAINER WHERE CITY NOT IN ("DELHI","MUMBAI");

Ans)

TID TNAME

103 DEEPTI

106 MANIPRABHA

(b) SELECT DISTINCT TID FROM COURSE;

Ans)DISTINCT TID

101

103

102

104

105

c) SELECT TID, COUNT(\*), MIN(FEES) FROM COURSE GROUP BY TID HAVING COUNT(\*)>1;

Ans)

TID COUNT(\*) MIN(FEES)

101 2 12000

d) SELECT COUNT(\*), SUM(FEES) FROM COURSE WHERE STARTDATE < '2018-09-15';

Ans: COUNT(\*) SUM(FEES)

4 65000

(e) SELECT MIN(STARTDATE) FROM COURSE;

Ans) MIN(STARTDATE)

2018-07-02

(f). SELECT MAX(STARTDATE), MIN(FEES) FROM COURSE;

Ans) max(StartDate) min(Fees)

2018-10-01 9000

(g) SELECT CITY, SUM(SALARY) FROM TRAINER GROUP BY CITY HAVING COUNT(\*)>1 ;

Ans) CITY SUM(SALARY)

MUMBAI 18,5000

DELHI 15,8000

2) Consider the following tables SCHOOL and ADMIN. Write SQL commands for the statements (i) to (iv) and give outputs for SQL queries (v) to (viii).

TABLE : SCHOOL

| ODE  | TEACHE<br>RNAME | SUBJECT       | DOJ        | PERI<br>ODS | EXP<br>ERI<br>ENC<br>E |
|------|-----------------|---------------|------------|-------------|------------------------|
| 1001 | RAVI<br>SHANKAR | ENGLISH       | 12/03/2000 | 24          | 10                     |
| 1009 | PRIYA<br>RAI    | PHYSICS       | 03/09/1998 | 26          | 12                     |
| 1203 | LISA<br>ANAND   | ENGLISH       | 09/04/2000 | 27          | 5                      |
| 1045 | YASHRAJ         | MATHS         | 24/08/2000 | 24          | 15                     |
| 1123 | GAMAM           | PHYSICS       | 16/07/1999 | 28          | 3                      |
| 1167 | HARISH B        | CHEMIST<br>RY | 19/10/1999 | 27          | 5                      |
| 1215 | UMESH           | PHYSICS       | 11/05/1998 | 22          | 16                     |

TABLE : ADMIN

| CODE | GENDER | DESIGNATION    |
|------|--------|----------------|
| 1001 | MALE   | VICE PRINCIPAL |
| 1009 | FEMALE | COORDINATOR    |
| 1203 | FEMALE | COORDINATOR    |
| 1045 | MALE   | HOD            |
| 1123 | MALE   | SENIOR TEACHER |
| 1167 | MALE   | SENIOR TEACHER |
| 1215 | MALE   | HOD            |

(i) To decrease period by 10% of the teachers of English subject.

UPDATE SCHOOL SET PERIOD = PERIOD\*0.90;

(ii) To display TEACHERNAME, CODE and DESIGNATION from tables SCHOOL and ADMIN whose gender is male.

SELECT S.TEACHERNAME,S.CODE, A.DESIGNATION FROM SCHOOL, ADMIN A WHERE GENDER="MALE" AND S.CODE=A.CODE;

(iii) To display number of teachers in each subject.

SELECT SUBJECT, COUNT(\*) FROM SCHOOL GROUP BY SUBJECT;

(iv) To display details of all teachers who have joined the school after 01/01/1999 in descending order of experience.

SELECT S.CODE,S.TEACHERNAME, S.SUBJECT, S.DOJ,S.PERIODS, S.EXPERIENCE,A.GENDER,A.DESIGNATION FROM SCHOOL S, ADMIN A WHERE DOB>"01/01/1999" AND S.CODE=A.CODE ORDER BY EXPERIENCE DESC;

(v) SELECT SUM(PERIODS), SUBJECT FROM SCHOOL GROUP BY SUBJECT;

| SUM(PERIODS) | SUBJECT   |
|--------------|-----------|
| 51           | ENGLISH   |
| 76           | PHYSICS   |
| 24           | MATHS     |
| 27           | CHEMISTRY |

(vi) SELECT TEACHERNAME, GENDER FROM SCHOOL, ADMIN WHERE DESIGNATION ="COORDINATOR" AND SCHOOL.CODE=ADMIN.CODE;

| TEACHERNAME | GENDER |
|-------------|--------|
| PRIYA RAI   | FEMALE |
| LISA ANAND  | FEMALE |

(vii) SELECT DESIGNATION, COUNT(\*) FROM ADMIN GROUP BY DESIGNATION HAVING COUNT(\*)>1;

| DESIGNATION    | COUNT(*) |
|----------------|----------|
| COORDINATOR    | 2        |
| HOD            | 2        |
| SENIOR TEACHER | 2        |

(viii) SELECT COUNT(DISTINCT SUBJECT) FROM SCHOOL; COUNT(\*)

3) Given the following Teacher Relation. Write SQL Commands from (a) to (g)

| No | Name        | Department  | DateofJoining | Salary | Sex |
|----|-------------|-------------|---------------|--------|-----|
| 1  | Raja        | Computer    | 21/5/98       | 8000   | M   |
| 2  | Sangita     | History     | 21/5/97       | 9000   | F   |
| 3  | Ritu        | Sociology   | 29/8/98       | 8000   | F   |
| 4  | Kumar       | Linguistics | 13/6/96       | 10000  | M   |
| 5  | Venkatraman | History     | 31/10/99      | 8000   | M   |
| 6  | Sindhu      | Computer    | 21/5/86       | 14000  | M   |
| 7  | Aishwarya   | Sociology   | 11/1/1998     | 12000  | F   |

(a) To select all the information of teacher in computer department

Ans: Select \* from Teacher where Department="Computer"

(b) To list the name of female teachers in History Department.

Ans: Select Name from Teacher Where Sex="F" And Department="History"

(c) To list all names of teachers with date of admission in ascending order.

Ans: Select Name from Teacher Order By Dateofjoining Asc

(d) To display Teacher's Name, Department, and Salary of female teachers

Ans: Select Name,Department,Salary from Teacher Where Sex="F"

(e) To count the number of items whose salary is less than 10000

Ans: Select Count(\*) from Teacher Where Salary<10000

(f) To insert a new record in the Teacher table with the following data: 8,"Mersha","Computer", (1/1/2000),12000,"M".

Ans: Insert into Teacher values ,"Mersha", "Computer", {1/1/2000},12000,"M");

=====XXXXXX=====



## Tips & Tricks for Score 100 marks in Computer Science.

- A student must go through the CBSE Class 12th syllabus for computer science before starting to prepare for the exam. The syllabus would also tell you the topics with a higher weightage of marks, which can not be skipped or avoided. It would also help the students in forming a proper study plan.
- A student must utilize the initial 15 minutes for reading the question paper.
- They must analyse all the answers they are confident about and must start with them.
- Try to write in bullet points and highlight the crucial points.
- Make a rough sketch for the programs first to avoid making mistakes while writing the final program.
- Always keep the last 15 minutes for revision and rectifying any mistakes.
- Do not stick to any question for too long if you don't know the answer.
- Be specific and neat in your answers.
- The key to a successful computer science exam is understanding. As most of the question paper is based on programming, the only way to score is by having a thorough knowledge of programming.
- Please create your notes for the essential terms and concepts after understanding them. The given notes in a book would be suitable for learning, but for revision, your messages will provide you with the exact information you need.
- Clear any doubts you have as soon as you can. Use your class 11th computer science books or reference books to clear your doubts, but the best solution would be to ask a teacher.
- Practice all the exercises and questions available in your textbook.
- Remember to study definitions and their basic applications thoroughly; questions often arise asking for explanations of topics.
- Create flowcharts to help you understand and remember programming.
- When you practice writing programs, try to maintain the required spacing, even if you answer only in rough. Getting into the habit of writing according to the proper spacing will help you.
- If you have time, use different colored pens/pencils for other areas to help you visually remember them.
- Pay attention during practical classes; this will help you score high in the applicable marks section worth 30 marks and help clear your doubts.
- Revise by practicing programming on a computer; you will immediately find out where you stand regarding how many programs you can successfully write/run.
- Pay attention to details; a single misplaced punctuation mark can ruin an entire answer.
- Practice sample papers/previous year papers to understand the pattern, test your preparation, and identify your weak areas.
- Remember always to correct the articles you attempt so you can find and improve your weak areas.
- No matter, if you choose to become a computer programmer or a doctor, basic knowledge of computer science, Will always be an added advantage. Use logic and understanding, and your class 12th computer science exam will be over before you know it.
- Using every resource available to you can help you be a more versatile student. Various Minimum learning material, HOTS, Different materials on KVS RO websites , Various blogs and especially previous years SEE as well as compartment exam paper.

XXXXXXXXXX