Chapter 1. Exception Handling in Python (1 mcq + 1 fib + 3*1 =5 Marks)

Syntax Errors: Syntax errors are detected when we have not followed the rules of the particular programming language while writing a program. These errors are also known as parsing errors.

Exceptions: An exception is a Python object that represents an error.

Built-in Exceptions: Commonly occurring exceptions are usually defined in the compiler/interpreter are called built-in exceptions.

Built-in exceptions that can be raised in Python:

1	SyntaxError	It is raised when there is an error in the syntax of the Python code. print “hello”
2	ValueError	It is raised when a built-in method or operation receives an argument that has the right data type but mismatched or inappropriate values. math.sqrt(-16)

3	ZeroDivisionError	It is raised when the denominator in a division operation is zero.
		a=10
		b=0
		print(a/b)

4	IOError	It is raised when the file specified in a program statement cannot be opened. file = open("abcd.txt","r")
5	KeyboardInterrupt	It is raised when the user accidentally hits the Delete or Esc key while executing a program due to which the normal flow of the program is interrupted.
6	ImportError	It is raised when the requested module definition is not found.
8	EOFError	It is raised when the end of file condition is reached without reading any data by input().
9	IndexError	It is raised when the index or subscript in a sequence is out of range.
10	NameError	It is raised when a local or global variable name is not defined.
11	IndentationError	It is raised due to incorrect indentation in the program code.
12	TypeError	It is raised when an operator is supplied with a value of incorrect data type.
12	OverFlowError	It is raised when the result of a calculation exceeds the maximum limit for numeric data type.

The raise Statement: The raise statement can be used to throw an exception.

Syntax: raise exception-name[(optional argument)]

Example:

```
numbers = [40,50,60,70]
length=10
if length>len(numbers):
    raise Exception("An exception has occurred")
else:
    print(length)
```

The assert Statement: An assert statement in Python is used to test an expression in the program code. The exception is raised, if the result after testing becomes false.

Syntax: assert Expression[,arguments]

Example:

```
def negativecheck(number):
        assert(number>=0),    "OOPS... Negative Number"
        print(number*number)
print (negativecheck(100))
print (negativecheck(-350))
```

Exception Handling: The process of writing additional code in a program to give proper messages to the user on encountering an exception is known as exception handling.

Exception object: When an error occurs, Python interpreter creates an object called the Exception object

Throwing an exception: The process of creating an exception object and handing it over to the runtime system is called throwing an exception.

Catching the exception: The process of executing a suitable exception handler is known as catching the exception.

Call stack: Call stack is the list of methods that are called one after another, used by the program to manage method execution and handle exceptions.

Catching Exceptions using try and except block:

- While writing or debugging a program, a user might doubt an exception to occur in a particular part of the code. Such suspicious lines of codes are put inside a try block.
- Exceptions, if any, are caught in the try block and handled in the except block.
- Every try block is followed by an except block.
- The appropriate code to handle each of the possible exceptions are written inside the except clause.
- If an exception is raised for which no handler is created by the programmer, then such an exception can be handled by adding an except clause without specifying any exception. This except clause should be added as the last clause of the try…except block.

Syntax:

```
try:
        [program statements where exceptions might occur]
except [exception-name]:
        [code for exception handling if the exception-name error is encountered]
```

Example:

```
print ("Practicing for try block")
try:
        numerator=50
        denom=int(input("Enter the denominator"))
        quotient=(numerator/denom)
        print(quotient)
        print ("Division performed successfully")
except ZeroDivisionError:
        print ("Denominator as ZERO.... not allowed")
        print("OUTSIDE try..except block")
except:
        print(" OOPS.....SOME EXCEPTION RAISED")
```

try...except…else clause:

- If there is no error then else clause will be executed.
- An except block will be executed only if some exception is raised in the try block.
- But if there is no error then none of the except blocks will be executed.
- In this case, the statements inside the else clause will be executed.

```
try:
        numerator=50
        denom=int(input("Enter the denominator: "))
        quotient=(numerator/denom)
        print ("Division performed successfully")
except ZeroDivisionError:
        print ("Denominator as ZERO is not allowed")
else:
        print ("The result of division operation is ", quotient)
```

Finally Clause:

- The statements inside the finally block are always executed regardless of whether an exception has occurred in the try block or not.
- The try statement in Python can also have an optional finally clause.
- It is a common practice to use finally clause while working with files to ensure that the file object is closed.

```
try:
        numerator=50
        denom=int(input("Enter the denominator: "))
        quotient=(numerator/denom)
        print ("Division performed successfully")
except ZeroDivisionError:
        print ("Denominator as ZERO is not allowed")
else:
        print ("The result of division operation is ", quotient)
finally:
        print ("OVER AND OUT")
```

Recovering and continuing with finally clause:

- If an error has been detected in the try block and the exception has been thrown, the appropriate except block will be executed to handle the error.
- But if the exception is not handled by any of the except clauses, then it is re-raised after the execution of the finally block.
- After execution of finally block, Python transfers the control to a previously entered try or to the next higher level default exception handler.

```
try:
        numerator=50
        denom=int(input("Enter the denominator"))
        quotient=(numerator/denom)
        print ("Division performed successfully")
except ZeroDivisionError:
        print ("Denominator as ZERO is not allowed")
else:
        print ("The result of division operation is ", quotient)
finally:
    print ("OVER AND OUT")
```

Chapter 2. File Handling in Python (1 mcq + 1 fib + 3*1 =5 Marks)

File: A file is a named location on a secondary storage media where data are permanently stored for later access.

Types of files:

There are mainly two types of data files - text file and binary file.

1. **Text file:**

- A text file can be understood as a sequence of characters consisting of alphabets, numbers and other special symbols.
- While opening a text file, the text editor translates each ASCII value and shows us the equivalent character that is readable by the human being.
- Each line of a text file is terminated by a special character, called the End of Line (EOL).
- When a text editor or a program interpreter encounters the ASCII equivalent of the EOL character, it displays the remaining file contents starting from a new line.

2. **Binary Files**

- Binary files are also stored in terms of bytes (0s and 1s), but unlike text files, these bytes do not represent the ASCII values of characters. Rather, they represent the actual content such as image, audio, video, compressed versions of other files, executable files, etc.
- These files are not human readable. Thus, trying to open a binary file using a text editor will show some garbage values.
- We need specific software to read or write the contents of a binary file.
- Binary files are stored in a computer in a sequence of bytes. Even a single bit change can corrupt the file and make it unreadable to the supporting application.

File opening methods / Different methods to open a file:

I. using open():

Syntax: file_object= open(file_name, access_mode)

Example: file1=open(“test.txt”, “r”)

II. Opening a file using with clause:

Syntax: with open (file_name, access_mode) as file_ object:

Example: with open(“myfile.txt”,”r”) as file1:

File access modes /File opening modes: myObject=open(“myfile.txt”, “a+”)

File Mode	Description
<r>	Opens the file in read-only mode.
<rb>	Opens the file in binary and read-only mode.
<r+> or <+r>	Opens the file in both read and write mode.
<w>	Opens the file in write mode. If the file already exists, all the contents will be overwritten. If the file doesn’t exist, then a new file will be created.
<wb+> or <+wb>	Opens the file in read, write and binary mode. If the file already exists, the contents will be overwritten. If the file doesn’t exist, then a new file will be created.
<a>	Opens the file in append mode. If the file doesn’t exist, then a new file will be created.
<a+> or <+a>	Opens the file in append and read mode. If the file doesn’t exist, then it will create a new file.

Closing a file: Once we are done with the read/write operations on a file, file need to be closed using close() method. While closing a file, the system frees the memory allocated to it. Syntax: file_object.close() Example: file1.close()

Writing to a text file:

1. **write():** write() takes a string as an argument and writes it to the text file. It returns the number of characters being written on single execution of the write().
   Also we need to add a newline character (\n) at the end of every sentence to mark the end of line. Syntax: fileobject.write(string) Example: myobject.write("Python")

2. **The writelines() method:** This method is used to write multiple strings to a file. We need to pass an iterable object like lists, tuple, etc. containing strings to the writelines() method.
   Syntax: fileobject.writelines(list1) Example: myobject.writelines(lines)

Reading from a text file:

1. **The read() method:** This method is used to read a specified number of bytes of data from a data file. Syntax: file_object.read(n) Example: lines=myobject.read(10)

2. **The readline([n]) method:** This method reads one complete line from a file where each line terminates with a newline (\n) character.
   Syntax: fileobject.readline() Example: line=myobject.readline()

3. **The readlines() method:** The method reads all the lines and returns the lines along with newline as a list of strings.
   Syntax : file_object readlines( ) Example: line=myobject.readlines())

Setting offsets in a File: To access data in a random fashion, then Python gives us seek() and tell() functions.

The tell() method:
This function returns an integer that specifies the current position of the file object in the file. Syntax: file_object.tell() Example: file1.tell()

The seek() method :
This method is used to position the file object at a particular position in a file.
Syntax: file_object.seek(offset , reference_point) Example: file1.seek(0)

Pickle: To save any object structure along with data, Python provides a module called Pickle.
Serialization (pickling): Serialization is the process of transforming data or an object in memory to a stream of bytes called byte streams.
De-serialization: De-serialization or unpickling is the inverse of pickling process where a byte stream is converted back to Python object

The module Pickle is used for serializing and de-serializing any Python object structure.

The pickle module provides two methods – dump() method, load() method

1. **The dump() method:** This method is used to convert (pickling) Python objects for writing data in a binary file.

Syntax: dump(data_object, file_object) Example: pickle.dump(list1,file1)

2. **The load() method :** This method is used to load (unpickling) data from a binary file.

Syntax: file_object = load(file_object) Example: list1=pickle.load(file1)

Chapter 3. Stack (1 mcq+2x1+5x1=8 Marks)

Stack:

- Arrangement of elements in a linear order is called a stack, where element inserted in the last will be removed first from the TOP end.
- Stack is a data structure in which insertion and deletion is done from one end only, usually referred to as TOP.
- Stack follows LIFO principle using which an element inserted in the last will be removed first.

Applications of Stack:

a) Some of the applications of stack in real-life are:

- Pile of clothes in an almirah.
- Multiple chairs in a vertical pile.
- Bangles worn on wrist.

- Pile of boxes of eatables in pantry or on a kitchen shelf.

b) Some examples of application of stack in programming:

- To reverse a string, its characters are traversed from the last to the first, which can be easily done by pushing the characters of the string into a stack.
- In text or image editors, the redo and undo options work by using a stack to keep track of changes, so when the redo or undo icon is clicked, the most recent editing action is redone or undone.
- While browsing the web, we move from one web page to another by accessing links between them. In order to go back to the last visited web page, we may use the back button on the browser. In this case, the history of browsed pages is maintained as stack.
- To handle matching of parentheses, stack is used.

Operations on Stack: PUSH and POP:

- **PUSH** adds a new element at the TOP of the stack. It is an insertion operation. We can add elements to a stack until it is full. A stack is full when no more elements can be added to it. Trying to add an element to a full stack results in an exception called 'overflow'.
- **POP** operation is used to remove the top most element of the stack, that is, the element at the TOP of the stack. It is a delete operation. We can delete elements from a stack until it is empty i.e. there is no element in it. Trying to delete an element from an empty stack results in an exception called 'underflow'.

Implementation of Stack in Python:

1. Create an empty stack named glassStack.

```
glassStack = list()
```

2.

```
def isEmpty(glassStack):
    if len(glassStack)==0:
        return True
    else:
        return False
```

3.

```
def Push(glassStack,element):
    glassStack.append(element)
```

4.

```
def Pop(glassStack):
    if isEmpty(glassStack):
        print('underflow')
        return None
    else:
        return(glassStack.pop())
```

5.

```
def size(glassStack):
    return len(glassStack)
```

6.

```
def top(glassStack):
    element=glassStack[len(glassStack)-1]
    return element
```

Infix, Prefix and Postfix Notations:

Type of Expression	**Description**	**Example**
Infix	Operators are placed in between the operands	x * y + z
Prefix(Polish)	Operators are placed before the corresponding operands	+xy
Postfix(Reverse Polish)	Operators are placed after the corresponding operands	xy+

Algorithm: Conversion of expression from infix to postfix notation:

```
Step 1: Create an empty string named postExp to store the converted postfix expression.
Step 2: INPUT infix expression in a variable, say inExp
Step 3: For each character in inExp, REPEAT Step 4
Step 4:
   a) IF character is a left parenthesis THEN PUSH it onto the Stack
   b) ELSE IF character is a right parenthesis THEN POP the elements from the Stack and
      append them to postExp until the corresponding left parenthesis is found. Discard both
      left and right parentheses.
   c) ELSE IF character is an operator
         i) IF its precedence is lower than the operator at the top of the Stack
            THEN POP operators from the Stack and append them to postExp utill
            an operator with lower precedence is found
         ii) ELSE PUSH operator on to the Stack.
   d) ELSE if the character is an operand, then directly append the character to postExp
Step 5: After scanning the complete infix expression, all remaining operators from the stack
and append to postExp until Stack is empty.
Step 6: OUTPUT the postfix expression postExp
```

Example: (X+Y) / (Z*8)

Symbol	**Stack**	**Postfix exp**
(	(	
X	(	X
+	(+	X
Y	(+	XY
)		XY+
/	/	XY+
(	/(	XY+
Z	/(	XY+Z
*	/(*	XY+Z
8	/(*	XY+Z8
)	/	XY+Z8*
	Null	XY+Z8*/

Example: (A + B / C * (D + C) – F)

Symbol	Stack	Postfix exp
(	(	
A	(	A
+	(+	A
B	(+	AB
/	(+/	AB
C	(+/	ABC
*	(+*	ABC/
(	(+*(	ABC/
D	(+*(	ABC/D
+	(+*(+	ABC/D
C	(+*(+	ABC/DC
)	(+*	ABC/DC+
-	(-	ABC/DC+*+
F	(-	ABC/DC+*+F
)		ABC/DC+*+F-

Algorithm: Evaluation of postfix expression:

Step 1: INPUT postfix expression in a variable, say postExp
Step 2: For each character in postExp, REPEAT Step 3
Step 3: IF character is an operand THEN PUSH character on the Stack
ELSE POP two elements from the Stack, apply the operator on the popped elements and PUSH the computed value onto the Stack
Step 4: IF Stack has a single element THEN POP the element and OUTPUT as the net result
ELSE
OUTPUT "Invaild Postfix expression"

Example: Evaluate the following postfix expression while showing status of stack after each operation given A=3, B=5, C=1,D=4

AB+C*=35 + 1 *

Symbol	Stack
3	3
5	35
+	3+5=8
1	81
*	8*1=8

AB*C/D* =35*1/4*

Symbol	Stack
3	3
5	35
*	3*5=15
1	15 1
/	15/1=15
4	15 4
*	15*4=60

Chapter 4. Queue (1 MCQ +2x1+5x1=8 Marks)

Queue:

Queue is an ordered linear list of elements, having different ends for adding and removing elements in it. The end used for adding the elements is called rear, and front end is used to remove the elements from the queue.

Queue data structure works on First-In-First-Out (FIFO) principle.

Examples of queue in our everyday life:

- Students standing in a queue for morning assembly.
- Customers forming a queue at the cash counter in a bank.
- Vehicles queued at fuel pumps.
- Queue of people at a bank.

First In First Out (FIFO): Queue follows the principle of First In First Out (FIFO), since the element entering first in the queue will be the first one to come out of it. It is also known as a First Come First Served (FCFS) approach.

Applications of Queue:

A. Applications in real-life:

- If a train ticket is in the waiting list (such as W/L1), it means the ticket is in a queue of tickets waiting to get confirmed, as per the increasing order of waiting numbers. If a confirmed ticket is cancelled, the W/ L1 numbered ticket is removed from the FRONT of the waiting queue and confirmed.
- Sometimes on calling a customer service center, the Interactive Voice Response System (IVRS) tells us to wait till a support person is available. Here the call is put into a queue of customers waiting to be serviced.
- Imagine there is a single-lane one-way road, and then the vehicle that entered first will exit first, following the concept of queue. Likewise, vehicles in a highway toll tax booth are served following the principle of FIFO.

B. Application of queue in computer science:

- Suppose there is a web-server hosting a web-site to declare results. This server can handle a maximum of 50 concurrent requests to view results. So, to serve thousands of user requests, a Queue would be the most appropriate data structure to use.
- In a multitasking operating system, jobs are queued and then given access to the processor according to some order.
- When we send print commands from multiple files from the same computer or from different computers using a shared printer. The OS puts these print requests in a queue and sends them to the printer one by one on a FIFO basis.

Operations on Queue:

1. ENQUEUE: Enqueue operation is used to insert a new element to the queue at the rear end. Inserting elements beyond capacity of the queue will result in an exception - known as Overflow.
2. DEQUEUE: Dequeue operation is used to remove one element at a time from the front of the queue. Trying to delete an element from an empty queue will result in exception - known as Underflow.
3. ISEMPTY: Isempty operation is used to check whether the queue has any element or not.
4. PEEK: Peek operation is used to view elements at the front of the queue, without removing it from the queue.
5. ISFULL: Isfull operation used to check whether any more elements can be added to the queue or not.

Show the status of queue after each operation:

enqueue(34)	F→34 ← R
enqueue(54)	F→34 54 ←R

dequeue()	F→ 54 ←R	
enqueue(12)	F→ 54 12←R	
dequeue()	F→ 12←R	
enqueue(61)	F→ 12 61←R	
peek()	RETURNS 12	F→ 12 61←R
dequeue()	F→ 61←R	
dequeue()	Null [Queue is Empty]	
dequeue()	Underflow	

Implementation of Queue using Python:

1. Let's create a queue named myQueue.   myQueue = list()

2.
```
def enqueue(myQueue, element):
    myQueue.append(element)
```

3.
```
def isEmpty(myQueue):
        if len(myQueue)==0:
                return True
        else:
                return False
```

4.
```
def dequeue(myQueue):
        if not (isEmpty(myQueue)):
                return myQueue.pop(0)
        else :
                print("Queue is empty")
```

5.
```
def size(myQueue):
        return len(myQueue)
```

6.
```
def peek(myQueue):
        if isEmpty(myQueue):
                print('Queue is empty')
                        return None
        else:
                return myQueue[0]
```

Introduction to Deque:

- Deque is an arrangement in which addition and removal of elements can happen from any end rear or front.
- It is also known as Double ended queue, because it permits insertion, deletion operations from any end.

Applications of Deque:

- At a train ticket purchasing counter, a normal queue of people is formed for purchasing a ticket. A person at the front purchased the ticket and left the counter. After a while they return back to the counter to ask something. As they have already purchased a ticket, they may have the privilege to join the queue from the front.

- Vehicles in a highway toll tax booth are served following the principle of queue. There are multiple queues if there are parallel booths at the toll gate. In case all vehicles of a booth are served then vehicles from the other booths are asked to form a queue in front of the vacant booth. So, vehicles at the end of those queues will leave current booth and join queue at the vacant booth.

Following are some examples where data structure deque may be applied in computer science:

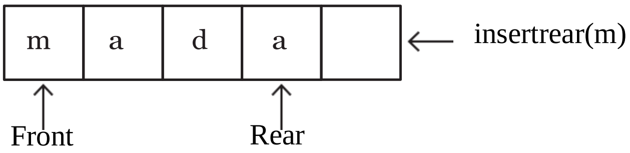
- To maintain browser history, usually a stack is used, because once a tab is closed and if you press ctrl+shift+T, the most recently closed URL is opened first. As the number of URLs which can be stored in history is fixed, so when this list of URLs becomes large, URLs from the end of the list gets deleted.
- For providing the Do and Undo option in any text editor.
- To check whether a given string is palindrome or not

Operations on Deque:

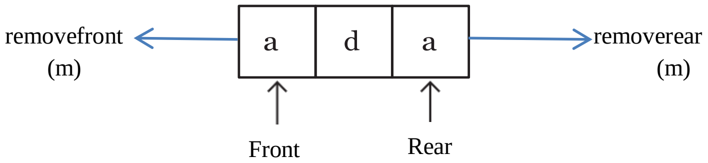
- INSERTFRONT: This operation is used to insert a new element at the front of the deque.
- INSERTREAR: This operation is the same as a normal queue, i.e. insert a new element at the rear of the deque.
- DELETIONFRONT: This operation is the same as normal queue, i.e. to remove an element from the front of the deque.
- DELETIONREAR: This operation is used to remove one element at a time from the rear of the deque.

Algorithm to check the string is palindrome or not using deque:

Step1: Start traversing string (madam) from left side, a character at a time.
Step 2: Insert the character in deque as normal queue using INSERTREAR.
Step 3: Repeat Step 1 and Step 2 for all characters of string (madam)



Step 4: Remove one character from the front and one character from the rear end of deque using DELETIONFRONT and DELETIONREAR we can do it.



Step 5: Match these two removed characters.
Step 6: If they are same then repeat Step 4 and 5 till deque is empty or left with only one character, eventually string is Palindrome else stop as string is not palindrome.

<u>Implementation of Deque Using Python:</u>

```
myDeque = list()
```

A function insertFront(), to insert an element at the front of deque:

```
def insertFront(myDeque, element):
        myDeque.insert(0,element)
```

A function insertRear(), to insert an element at the rear of deque:

```
def insertRear(myDeque, element):
   myQueue.append(element)
```

A function isEmpty(), to check the presence of element(s):

```
def isEmpty(myDeque):
   if len(myDeque)==0:
      return True
   else:
      return False
```

A function deletionRear(), to delete an element from the rear of the deque:

```
def deletionRear(myDeque):
         if not (isEmpty()):
                  return myDeque.pop()
         else :
                  print(“Deque empty”)
```

A function deletionFront(), to delete an element from the front of deque :

```
def dequeue(myDeque):
   if not (isEmpty(myDeque)):
      return myDeque.pop(0)
   else :
      print("Queue is empty")
```

Show the status of deque after each operation

peek()	Empty
insertFront(12)	F→12←
insertRear(67)	F→12 67←
deletionFront()	F→ 67←
insertRear(43)	F→ 67 43←
deletionRear()	F→ 67 ←
deletionFront()	Empty
deletionRear()	Underflow

- Elements ‘A’,’S’,’D’ and ‘F’ are present in the queue, and they are deleted one at a time, **A,S,D,F** is the sequence of element received.
- **DEQUE** is a data structure where elements can be added or removed at either end, but not in the middle.
- A deque contains ‘z’,’x’,’c’,’v’ and ‘b’. Elements received after deletion are ‘z’,’b’,’v’,’x’ and ‘c’. Following is the sequence of deletion operation performed on deque.

DELETIONFRONT()
DELETIONREAR()
DELETIONREAR()
DELETIONFRONT ()
DELETIONFRONT()

Chapter 5. Sorting (2 mcq+3x1+(HOT)5x1=10 Marks)

- Sorting is the process of ordering or arranging a given collection of elements in some particular order. **Types of sorting:** Bubble sort, Selection sort, Insertion sort

Bubble sort:

- The first sorting is Bubble sort. It **sorts a given list of elements by repeatedly comparing the adjacent elements and swapping them if they are unordered.**
- Swapping two elements means changing their positions with each other.
- For a list with n elements, the bubble sort makes a total of n – 1 passes to sort the list.

BUBBLESORT(numList, n)

```
Step 1: SET i = 0
Step 2: WHILE i< n REPEAT STEPS 3 to 8
Step 3:     SET j = 0
Step 4:         WHILE j< n-i-1,REPEAT STEPS 5 to 7
Step 5:             IF numList[j] > numList[j+1] THEN
Step 6:                 swap(numList[j],numList[j+1])
Step 7:             SET j=j+1
Step 8:  SET i=i+1
```

Tracing Example:

numlist	8	7	13	1	-9	4

Pass1						
8>7 So interchange	8	7	13	1	-9	4
8<13 No swapping	7	8	13	1	-9	4
13>1 So interchange	7	8	13	1	-9	4
13>-9 So interchange	7	8	1	13	-9	4
13>4 So interchange	7	8	1	-9	13	4
After pass1	7	8	1	-9	4	13
Pass2						
7<8 No swapping	7	8	1	-9	4	13
8>1 So interchange	7	8	1	-9	4	13
8>-9 So interchange	7	1	8	-9	4	13
8>4 So interchange	7	1	-9	8	4	13
After pass2	7	1	-9	4	8	13
Pass3						
7>1 So interchange	7	1	-9	4	8	13
7>-9 So interchange	1	7	-9	4	8	13
7>4 So interchange	1	-9	7	4	8	13

After pass3	1	-9	4	7	8	13
Pass4						
1>-9 So interchange	-9	1	4	7	8	13
1<4 No swapping	-9	1	4	7	8	13
After pass4	-9	1	4	7	8	13
Pass5						
-9<1 No swapping	-9	1	4	7	8	13
After pass5	-9	1	4	7	8	13

Final sorted list= -9, 1, 4, 7, 8, 13

SELECTION SORT:

- In Selection sort having n elements, it makes (n-1) number of passes through the list.
- The list is considered to be divided into two lists. The left list containing the sorted elements, and the right list containing the unsorted elements.
- Initially, the left list is empty, and the right list contains all the elements.
- In each iteration, **it finds the smallest element in the unsorted part and swaps it with the leftmost element in the unsorted part.** This process continues until the entire list is sorted.

SELECTIONSORT(numList, n)

```
Step 1: SET i=0
Step 2: WHILE i< n REPEAT STEPS 3 to 12
Step 3: SET min = i, flag = 0
Step 4: SET j= i+1
Step 5: WHILE j<n, REPEAT STEPS 6 to 9
Step 6: IF numList [j] < numList [min] THEN
Step 7: min = j
Step 8: flag = 1
Step 9: SET j=j+1
Step 10: IF flag = 1 THEN
Step 11:  swap(numList[i],numList[min])
Step 12:  SET i=i+1
```

Tracing Example:

8	7	13	1	-9	4

Pass1							
min= 8	8	7	13	1	-9	4	
	0	1	2	3	4	5	index
Compare 7 and 8, 7<8, min=7	8	7	13	1	-9	4	
Compare 13 and 7, 13>7, min=7	8	7	13	1	-9	4	

Compare 1 and 7, 1<7, min=7	8	7	13	1	-9	4
Compare -9 and 1, -9<1, min=-9	8	7	13	1	-9	4
Compare -9 and 4, 4>9, min=-9	8	7	13	1	-9	4
Swap min(-9) with first element(8)	-9	7	13	1	8	4
Pass2						
min=7	-9	7	13	1	8	4
Compare 7 and 13, 13>7, min=7	-9	7	13	1	8	4
Compare 1 and 7, 1<7, min=1	-9	7	13	1	8	4
Compare 1 and 8, 8>1, min=1	-9	7	13	1	8	4
Compare 1 and 4, 4>1, min=1	-9	7	13	1	8	4
Swap min(1) with second element(7)	-9	1	13	7	8	4
Pass3						
Min=13	-9	1	13	7	8	4
Compare 7 and 13, 7<13, min=7	-9	1	13	7	8	4
Compare 7 and 8, 7<8, min=7	-9	1	13	7	8	4
Compare 7 and 4, 4<7, min=4	-9	1	13	7	8	4
Swap min(4) with third element(13)	-9	1	4	7	8	13
Pass4						
Min=7	-9	1	4	7	8	13
Compare 7 and 8, 7<8, min=7	-9	1	4	7	8	13
Compare 7 and 13, 7<13, min=7	-9	1	4	7	8	13
No swap						
Min=8	-9	1	4	7	8	13
Compare 8 and 13, 8<13, min=8	-9	1	4	7	8	13
No Swap	-9	1	4	7	8	13

Final sorted list= -9, 1, 4, 7, 8, 13

Insertion Sort:

- Like Selection sort, in Insertion sort also, the **list is divided into two parts - one of sorted elements and another of unsorted elements.**
- Each element in the unsorted list is considered one by one and is inserted into the sorted list at its appropriate position.
- In each pass, **the sorted list is traversed from the backward direction to find the**

position where the unsorted element could be inserted. This sorting method is called insertion sort.

INSERTIONSORT(numList, n)

```
Step 1: SET i=1
Step 2: WHILE i< n REPEAT STEPS 3 to 9
Step 3: temp = numList[i]
Step 4: SET j = i-1
Step 5: WHILE j> = 0 and numList[j]>temp, REPEAT STEPS 6 to 7
Step 6: numList[j+1] = numList[j]
Step 7: SET j=j-1
Step 8: numList[j+1] = temp
Step 9:  SET i=i+1
```

Tracing Example:

8	7	13	1	-9	4

Pass1						
7<8, So swap	8	7	13	1	-9	4
After pass 1	7	8	13	1	-9	4
Pass2						
13>8, no swap	7	8	13	1	-9	4
After pass 2	7	8	13	1	-9	4
Pass3						
1<13, so swap	7	8	13	1	-9	4
1<8, so swap	7	8	1	13	-9	4
1<7, so swap	7	1	8	13	-9	4
After pass 3	1	7	8	13	-9	4
Pass4						
-9<13, so swap	1	7	8	13	-9	4
-9<8, so swap	1	7	8	-9	13	4
-9<7, so swap	1	7	-9	8	13	4
-9>1, so swap	1	-9	7	8	13	4
After pass4	-9	1	7	8	13	4
Pass5						
4<13, so swap	-9	1	7	8	13	4
8<4, so swap	-9	1	7	8	4	13
4<7, so swap	-9	1	7	4	8	13

1<4, no swap	-9	1	4	7	8	13
After pass5	-9	1	4	7	8	13

Final sorted list= -9, 1, 4, 7, 8, 13

Time Complexity of Algorithms: The amount of time an algorithm takes to process a given data can be called its time complexity.

Constant time algorithms: Any algorithm **that does not have any loop** will have time complexity as 1. Such algorithms are known as Constant time algorithms.

Linear time algorithms: Any algorithm that has **a loop** will have the time complexity as **n.** Such algorithms are known as Linear time algorithms.

Quadratic time algorithms: A loop within a **loop (nested loop)** will have the time complexity as n^2. Such algorithms are known as **Quadratic time algorithms**.

Chapter 6. Searching (2x1+3x1+(LOT)5x1=10 Marks)

Searching means locating a particular element in a collection of elements.

Linear Search (sequential search or serial search):

- Linear searching is a technique where every element of a given list is compared with the item to be searched (usually referred to as 'key').
- So, each element in the list is compared one by one with the key.
- This process continues until an element matching the key is found and we declare that the search is successful.
- If no element matches the key and we have traversed the entire list, we declare the search is unsuccessful i.e., the key is not present in the list.

Algorithm: Linear Search LinearSearch(numList, key, n)

```
Step 1: set index = 0
Step 2: while  index < n, repeat step 3
Step 3: if numlist[index]= key then
                print "Element found at position", index+1
                stop
        else
                index = index+1
Step 4: print "Search unsuccessful"
```

Example: [8, -4, 7, 17, 0, 2, 19], key=17.

index	**index < n**	**numList[index]= key**	**index=index+1**
0	0 < 7 ? Yes	8 = 17? No	1
1	1 < 7 ? Yes	-4 = 17 ? No	2
2	2 < 7 ? Yes	7 = 17 ? No	3
3	3 < 7 ? Yes	17 = 17 ? Yes	4

Output: Element found at position 4 (index=3).

Binary Search Method:

- For numeric values, the elements in the list may be arranged either in ascending or descending order of their key values.

- For textual data, it may be arranged alphabetically starting from a to z or from z to a.
- In binary search, the key to be searched is compared with the element in the middle of a sorted list. This could result in either of the three possibilities:
  - i) The element at the middle position itself matches the key or
  - ii) The element at the middle position is greater than the key or
  - iii) The element at the middle position is smaller than the key
  - o If the element at the middle position matches the key, we declare the search successful and the searching process ends.
  - o If the middle element is greater than the key, it must surely be in the first half. We can thus straight away ignore the second half of the list and repeat the searching process only in the first half.
  - o If the middle element is less than the key, it must be in the second half. We can thus straight away ignore the first half of the list and repeat the searching process only in the second half.
  - o This splitting and reduction in list size continued till the key is either found or the remaining list consists of only one item.
  - o If that item is not the key, then the search is unsuccessful.

Algorithm : Binary Search BinarySearch(numList, key)

```
Step 1: set first = 0, last = n-1
Step 2: while  first <= last repeat step 3 and 4
Step 3: calculate mid = (first+last)//2
Step 4: if numlist[mid] = key
                print “Element found at  position”, mid+1
                stop
        else if numlist[mid]>key, then
                last = mid 1 1
        else
                first=mid+1
Step 5: print “Search unsuccessful”
```

Example: numList = [2,3,5,7,10,11,12,17,19,23,29,31,3 7,41,43], key = 2

	First	last	Mid	numlist[mid]=key	key>numlist[mid]	first<=last
At Start	0	14	(0+14)//2=7	Not Known	Not Known	True
Iteration 1	0	14	(0+14)//2=7	17=2? No	2<17?True	0<=14? True
Iteration 2	0	6	(0+6)//2=3	7=2 ? No	2<7 ? True	0<=6? True
Iteration 3	0	2	(0+2)//2=1	3=2 ? No	2<3 ? True	0<=2? True
Iteration 4	0	0	(0+0)//2=0	2=2 ? Yes	Key found, Search terminates, return position as (mid+1)=1	

Applications of Binary Search:

- Searching a dictionary or telephone directory
- Finding the element with minimum value or maximum value in a sorted list

- Indexing in databases.
- Implementing routing tables in routers, data compression code.

Search By hashing:

- (Hashing is a technique which can be used to know the presence of a key in a list in just one step.)
- A formula called hash function is used to calculate the value at an index in the list.
- A hash function takes elements of a list one by one and generates an index value for every element.
- This will generate a new list called the hash table.
- Each index of the hash table can hold only one item and the positions are indexed by integer values starting from 0.
- Note that the size of the hash table can be larger than the size of the list.
- A simple hash function that works with numeric values is known as the remainder method. It takes an element from a list and divides it by the size of the hash table. The remainder so generated is called the hash value. h(element) = element % size(hash table)

Example: Consider a list of numbers 34, 16, 2, 93, 80, 77, 51. key=16

1. We can easily implement a hash table using a Python's List. Let us consider an empty hash table having 10 positions

Index	0	1	2	3	4	5	6	7	8	9	10
Value	None	None	None	None	None	None	None	None	None	None	None

2. We can use the hash function remainder method to create a hash table.

Element	34	16	2	93	80	77	51
Hash Value	34%10=4	16%10=6	2%10=2	93%10=3	80%10=0	77%10=7	51%10=1

3. After computing the hash values, each element is inserted at its designated position in the hash table shown below:

Index	0	1	2	3	4	5	6	7	8	9
Value	80	51	2	93	34	None	16	77	None	None

Result=Key 16 is found in the list at index 6.

- **Collision in hashing:** If two or more elements are in the same position in the list. This situation is called collision in hashing.
- **Collision resolution:** We must have a mechanism for placing the other items with the same hash value in the hash table. This process is called collision resolution.
- **Perfect hash function**: If every item of the list maps to a unique index in the hash table, the hash function is called a perfect hash function. If a hash function is perfect, collision will never occur.

Chapter 7. Understanding Data ((HOT)5x1=5 Marks)

The marks obtained by students in a subject are: 45, 50, 55, 60, 65, 70, 75, 80, 85, 90

Find: a) mean b) median c) mode d) range e) standard deviation

a) mean=$\frac{\text{sum}}{\text{n}}$

Sum = 45 + 50 + 55 + 60 + 65 + 70 + 75 + 80 + 85 + 90 = 675, n=10

Mean = 675 / 10= **67.5**

b) Median: Since there are 10 numbers (even),

Median = (5th value + 6th value) / 2
Median = (65 + 70) / 2= **67.5**

c) Mode: No mode (all numbers appear only once)

d) Range: Highest value − Lowest value

Range = 90 – 45=**45**

e) Standard deviation:

a. Calculate the Mean μ = $\frac{\text{sum of all values}}{n}$

$$\mu = \frac{45 + 50 + 55 + 60 + 65 + 70 + 75 + 80 + 85 + 90}{10}$$

μ =67.5

b. For each number x, find (x- μ) and square each difference $(x- \mu)^2$

Data (x)	Deviation (x−μ)	Squared Deviation (x−μ)2
45	45−67.5=−22.5	(−22.5)2=506.25
50	50−67.5=−17.5	(−17.5)2=306.25
55	55−67.5=−12.5	(−12.5)2=156.25
60	60−67.5=−7.5	(−7.5)2=56.25
65	65−67.5=−2.5	(−2.5)2=6.25
70	70−67.5=2.5	(2.5)2=6.25
75	75−67.5=7.5	(7.5)2=56.25
80	80−67.5=12.5	(12.5)2=156.25
85	85−67.5=17.5	(17.5)2=306.25
90	90−67.5=22.5	(22.5)2=506.25

c. Sum the Squared Deviations**:** Add all the values in the third column

$\sum(x - \mu)^2$ =$(506.25 + 306.25 + 156.25 + 56.25 + 6.25 + 6.25 + 56.25 + 156.25 + 306.25 + 506.25)^2$
= 2062.5

d. Calculate the Variance σ^2**:** Divide the sum by the number of data points

$\boldsymbol{\sigma^2 = \frac{2062.5}{10} = 206.25}$

e. **Calculate the Standard Deviationσ**: Take the square root of the variance.

σ= $\sqrt{206.25} \approx 14.3614$

Chapter 8. Database Concepts(1 mcq+1FIB+2x1+3x1+(LOT)5x1=12 Marks)

File System:

- File is a container to store data in a computer in the storage device of a computer system.

Limitations of a File System:

1. Difficulty in Access:

- Files themselves do not provide any mechanism to retrieve data.
- Sometimes it is difficult to access data in the required format and one has to write application program to access data.

2. Data Redundancy:

- Redundancy means same data are duplicated in different files.
- Redundancy leads to excess storage use and may cause data inconsistency also.

3. Data Inconsistency:

- Data inconsistency occurs when same data maintained in different places do not match.
- For example: If a student wants to get changed the spelling of her name, it needs to be changed in SName column in both the files (Student file and Attendance file).

4. Data Isolation:

- Data isolation occurs when there is no link or mapping between the files.
- The school will have to write separate programs to access different files.

5. Data Dependence:

- Data are stored in a specific format or structure in a file.
- If the structure or format itself is changed, all the existing application programs accessing that file also need to be changed.

6. Controlled Data Sharing:

- Every user should not be able to access all the data.
- The users should be given limited access to the file.

Database Management System

- A database management system (DBMS) is software that can be used to create and manage databases.
- Some examples of open source and commercial DBMS include MySQL, Oracle, MS SQL Server.

Retrieving data from a database through querying the database.

Use of Database in Real-life Applications:

Application	**Database to maintain data about**
Banking	customer information, account details, loan details, transaction details, etc.
Crop Loan	kisan credit card data, farmer’s personal data, land area and cultivation data, loan history, repayment data, etc
Inventory Management	product details, customer information, order details, delivery data, etc.
Organisation Resource Management	employee records, salary details, department information, branch locations, etc

Online Shopping	items description, user login details, users preferences details, etc

Key Concepts in DBMS:

1. **Database Schema:** Database schema is also called the visual or logical architecture as it tells us how the data are organized in a database.
2. **Data Constraint:** Certain restrictions or limitations on the type of data that can be inserted in one or more columns of a table are called constraints.
3. **Meta-data or Data Dictionary:** The database schema along with various constraints on the data is stored by DBMS in a database catalog or dictionary, called meta-data. A meta-data is data about the data.
4. **Database Instance**: Data stored in the database at a particular time.
5. **Query:** A query is a request to a database for obtaining information in a desired way.
6. **Data Manipulation:** It is the process of adding, updating, deleting, or retrieving data from a database.
7. **Database Engine:** Database engine is the set of programs used by a DBMS to create database and handle various queries for data retrieval and manipulation.

Relational Data Model:

- A data model describes the structure of the database, including how data are defined and represented, relationships among data, and the constraints.
- The most commonly used data model is Relational Data Model.
- Other types of data models include object-oriented data model, entity-relationship data model, document model and hierarchical data model.
- In relational model, tables are called relations that store data for different columns. Each table can have multiple columns where each column name should be unique.

Commonly used terminologies in relational data model:

- **ATTRIBUTE (FIELD)**: The columns of a table are the attributes or fields.
- **TUPLE**: Each row of data in a table is called a tuple.
- **DOMAIN**: It is a set of values from which an attribute can take a value in each row.
- **DEGREE**: The number of attributes in a relation.
- **CARDINALITY**: The number of tuples in a relation.

Three Important Properties of a Relation:

Property 1: imposes following rules on an attribute (column) of the relation.

- Each attribute in a relation has a unique name.
- Sequence of attributes in a relation is immaterial.

Property 2: governs following rules on a tuple (row) of a relation.

- Each tuple in a relation is distinct.
- Sequence of tuples in a relation is immaterial. The tuples are not considered to be ordered, even though they appear to be in tabular form.

Property 3: imposes following rules on the state of a relation.

- All data values in an attribute must be from the same domain.
- Each data value associated with an attribute must be atomic

- No attribute can have many data values in one tuple.
- A special value "NULL" is used to represent values that are unknown.

Keys in a Relational Database:

Candidate Key: It is a column or set of columns that can uniquely identify every record in a table**.**

Primary Key: It is a column that uniquely identifies the tuples in a relation is called the primary key.

Composite Primary Key: If no single attribute in a relation is able to uniquely distinguish the tuples, then more than one attribute are taken together as primary key. Such primary key consisting of more than one attribute is called Composite Primary key.

Foreign Key: A Foreign Key is an attribute whose value is derived from the primary key of another relation. It is used to represent relationship between two tables.

Chapter 9- Structured Query Language

(3 mcq+2FIB+2x1+3x1+(LOT)5x1=15 Marks)

RDBMS stands for Relational Database Management System. There are many RDBMS such as MySQL, Microsoft SQL Server, PostgreSQL, Oracle, etc.

Database is a collection of related tables. MySQL is a 'relational' DBMS.

Structured Query Language (SQL):

The Structured Query Language (SQL) is the most popular query language used by major relational database management systems such as MySQL, ORACLE, SQL Server, etc. SQL is not case sensitive.

Benefits of using SQL:

1. We do not have to specify how to get the data from the database. Rather, we simply specify what is to be retrieved, and SQL does the rest.
2. SQL provides statements for defining the structure of the data, manipulating data in the database, declaring constraints and retrieving data from the database in various ways, depending on our requirements.

Data Types and Constraints in MySQL:

Data type	**Description**
CHAR(n)	Specifies character type data of length n where n could be any value from 0 to 255**. CHAR is of fixed length, means, declaring CHAR (10) implies to reserve spaces for 10 characters. If data does not have 10 characters (e.g., 'city' has four characters), MySQL fills the remaining 6 characters with spaces padded on the right.**
VARCHAR(n)	Specifies character type data of length where n could be any value from 0 to 65535. But **unlike CHAR, VARCHAR (n) is a variable-length data type.** That is, declaring VARCHAR (30) means a maximum of 30 characters can be stored but the actual allocated bytes will depend on the length of entered string. So 'city' in VARCHAR (30) will occupy space needed to store 4 characters only.

INT	INT specifies an integer value. Each INT value occupies 4 bytes of storage. The range of unsigned values allowed in a 4 byte integer type are 0 to 4,294,967,295. For values larger than that, we have to use BIGINT, which occupies 8 bytes.
FLOAT	Holds numbers with decimal points. Each FLOAT value occupies 4 bytes.
DATE	The DATE type is used for dates in 'YYYY-MM-DD' format. YYYY is the 4 digit year, MM is the 2 digit month and DD is the 2 digit date. The supported range is '1000-01-01' to '9999-12-31'.

Constraints: Constraints are the certain types of restrictions on the data values that an attribute can have. They are used to ensure correctness of data.

Constraint	**Description**
NOT NULL	Ensures that a column cannot have NULL values where NULL means missing/ unknown/not applicable value.
UNIQUE	Ensures that all the values in a column are distinct/unique
DEFAULT	A default value specified for the column if no value is provided
PRIMARY KEY	The column which can uniquely identify each row/record in a table.
FOREIGN KEY	The column which refers to value of an attribute defined as primary key in another table

Schema: Schema includes creating a relation and giving name to a relation, identifying the attributes in a relation, deciding upon the data type for each attribute and also specifies the constraints as per the requirements.

SQL Commands:
DDL (Data Definition Language) - Create, Alter and Drop.
DML (Data Manipulation Language Commands)- insert, select, update and delete.

A. DDL Commands:

1. CREATE: The Create statement is used to create a database and its tables.

Syntax: CREATE DATABASE databasename;
Example: CREATE DATABASE StudentAttendance;

To know the names of existing databases, we use the statement **SHOW DATABASES**;

To use the StudentAttendance database: **USE StudentAttendance;**

To list the names of all the tables within a database: **SHOW TABLES;**

SQL Commands:

```
CREATE TABLE STUDENT(
                                RollNumber INT,
                                SName VARCHAR(20),
SDateofBirth DATE,
GUID CHAR (12),
PRIMARY KEY (RollNumber));
```

Describe Table: We can view the structure of an already created table using the DESCRIBE statement or DESC statement.

Syntax: DESCRIBE tablename;
Example: DESCRIBE STUDENT;

2. **ALTER Command:** ALTER TABLE statement is used to make changes in the structure of a table like adding, removing or changing data type of the existing columns or to add constraint in attribute.

a. Add primary key to a relation:

ALTER TABLE GUARDIAN ADD PRIMARY KEY (GUID);
ALTER TABLE ATTENDANCE ADD PRIMARY KEY(AttendanceDate, RollNumber);

b. Add foreign key to a relation:

ALTER TABLE STUDENT ADD FOREIGN KEY(GUID) REFERENCES GUARDIAN(GUID);

c. Add constraint UNIQUE to an existing attribute:

ALTER TABLE GUARDIAN ADD UNIQUE(GPhone);

d. Add an attribute to an existing table:

ALTER TABLE GUARDIAN ADD income INT;

e. Modify data type of an attribute:

ALTER TABLE table_name MODIFY attribute DATATYPE;

f. Modify constraint of an attribute:

ALTER TABLE STUDENT MODIFY SName VARCHAR(20) NOT NULL;

g. Add default value to an attribute:

ALTER TABLE STUDENT MODIFY SDateofBirth DATE DEFAULT ‘2000-05- 15’;

h. Remove an attribute:

ALTER TABLE GUARDIAN DROP income;

i. Remove primary key from the table:

ALTER TABLE GUARDIAN DROP PRIMARY KEY;

3. **DROP Command:** DROP statement is used to remove a database or a table permanently from the system.

DROP TABLE table_name;
DROP DATABASE database_name;

B. **DML Commands: (SQL for Data Manipulation):** Data Manipulation using a database means either insertion of new data, removal of existing data or modification of existing data in the database.

1. **INSERT Command:** INSERT INTO statement is used to insert new records in a table.

Syntax: INSERT INTO tablename VALUES (value 1, value 2,....);

Example 1: INSERT INTO GUARDIAN VALUES (444444444444, 'Amit Ahuja', 5711492685, 'G-35,Ashok vihar, Delhi');

2. **SELECT Command:** The SQL statement SELECT is used to retrieve data from the tables in a database and is also called a query statement.

Syntax:

SELECT attribute1, attribute2, ...
FROM table_name
WHERE condition;

Example:

a. **Retrieve selected columns:** SELECT EmpNo, Ename FROM EMPLOYEE;

b. **Renaming of columns:** SELECT EName **as Name** FROM EMPLOYEE;

c. **Distinct Clause:** Returns records without repetition.
SELECT DISTINCT CITY FROM STUDENT;

d. **Where clause:** The WHERE clause is used to retrieve data that meet some specified conditions.

SELECT Salary FROM EMPLOYEE WHERE EMPNO=101;

e. **Select records of all the employees except Aaliya.**
SELECT * FROM EMPLOYEE WHERE NOT Ename = 'Aaliya';

f. **select the name and department number of all those employees who are earning salary between 20000 and 50000:**
SELECT Ename, DeptId FROM EMPLOYEE
WHERE Salary BETWEEN 20000 AND 50000;

g. **Select details of all the employees who work in the departments having deptid D01, D02 or D04:**
SELECT * FROM EMPLOYEE
WHERE DeptId = 'D01' OR DeptId = 'D02' OR DeptId = 'D04';

h. **Membership operator IN:**

SELECT * FROM EMPLOYEE WHERE DeptId IN ('D01', 'D02' , 'D04');

i. **Select details of all the employees except those working in department number D01or D02.**
SELECT * FROM EMPLOYEE WHERE DeptId NOT IN('D01', 'D02');

- **ORDER BY Clause:** ORDER BY clause is used to display data in an ordered form with respect to a specified column.
- By default, ORDER BY displays records in ascending order.
- To display the records in descending order, the DESC keyword needs to be written with that column.

Select details of all the employees in ascending order of their salaries:

```
SELECT * FROM EMPLOYEE
ORDER BY Salary;
```

Select details of all the employees in descending order of their salaries:

```
SELECT * FROM EMPLOYEE
ORDER BY Salary DESC;
```

Handling NULL Values:

- SQL supports a special value called NULL to represent a missing or unknown value.
- NULL is used to represent missing or unknown values.
- NULL is different from zero.
- Any arithmetic operation performed with NULL value gives NULL.
- For example: **5 + NULL = NULL** because NULL is unknown hence the result is also unknown.

Select details of all those employees who have not been given a bonus:

```
SELECT * FROM EMPLOYEE WHERE Bonus IS NULL;
```

Select names of all employees who have been given a bonus (i.e., Bonus is not null) and works in the department D01:

```
SELECT EName FROM EMPLOYEE WHERE Bonus IS NOT NULL AND DeptID =
‘D01’;
```

Substring pattern matching:

SQL provides a LIKE operator that can be used with the WHERE clause to search for a specified pattern in a column.

The LIKE operator makes use of the following two wild card characters:
% (per cent) - used to represent zero, one, or multiple characters.
_ (underscore) - used to represent exactly a single character.

Select details of all those employees whose name starts with 'K':

```
SELECT * FROM EMPLOYEE WHERE Ename like 'K%';
```

Select details of all those employees whose name ends with 'a', and gets a salary more than 45000:

```
SELECT * FROM EMPLOYEE WHERE Ename like '%a' AND Salary > 45000;
```

Select details of all those employees whose name consist of exactly 5 letters and start with any letter but has ‘ANYA’ after that:

```
SELECT * FROM EMPLOYEE WHERE Ename like '_ANYA';
```

Select names of all employees containing 'se' as a substring in name:

```
SELECT Ename FROM EMPLOYEE WHERE Ename like '%se%';
```

Select names of all employees containing 'a' as the second character:

```
SELECT EName FROM EMPLOYEE WHERE Ename like '_a%';
```

3. **Update Command:** The UPDATE statement is used to make such modifications in existing data.

Syntax:

```
UPDATE table_name
SET attribute1 = value1, attribute2 = value2, ...
WHERE condition;
```

Example:

```
UPDATE STUDENT
SET GUID = 101
WHERE RollNumber = 3;
```

4. **Delete command:** DELETE statement is used to delete/remove one or more records from a table.

Syntax:

```
DELETE FROM table_name
WHERE condition;
```

Example:

```
DELETE FROM STUDENT WHERE RollNumber = 2;
```

Functions in SQL:

Single Row Functions: (Scalar Functions)

Single row functions are applied on a single value and return a single value.

Single row functions under three categories -

- Numeric (Math) Functions: POWER (). ROUND(), MOD(),
- String Functions- UCASE(), LCASE(), MID(), LENGTH(), LEFT(), RIGHT(), INSERT(), LTRIM(), RTRIM(), TRIM()
- Date and Time-NOW(), DATE(), MONTH(), MONTHNAME(), YEAR(), DAY()

A. Math Functions:

Function	Description	Example with output
POWER(X,Y) can also be written as POW(X,Y)	Calculates X to the power Y.	SELECT POWER(2,3); Output: 8

ROUND(N,D)	Rounds off number N to D number of decimal places.
	Note: If D=0, then it rounds off the number to the nearest integer.
		SELECT ROUND(2912.564, 1);
		Output: 2912.6
		SELECT ROUND(283.2);
		Output: 283

MOD(A, B)	Returns the remain der after dividing number A by number B.	SELECT MOD(21, 2); Output: 1

B. String Functions:

Function	Description	Example with output
UCASE(string) OR UPPER(string)	Converts string into uppercase.	SELECT UCASE(“Informatics Practices”);
LOWER(string) OR LCASE(string)	Converts string into lowercase.	SELECT LOWER(“Informatics Practices”);
MID(string, pos, n)	Returns a substring of size n starting from the specified position (pos) of the string. If n is not specified, it returns the substring from the position pos till end of the string.	SELECT MID(“Informatics”, 3, 4);
LENGTH(string)	Return the number of characters in the specified string.	SELECT LENGTH(“Informatics”);
LEFT(string, N)	Returns N number of characters from the left side of the string.	SELECT LEFT(“Computer”, 4);
RIGHT(string, N)	Returns N number of characters from the right side of the string.	SELECT RIGHT(“SCIENCE”, 3);
INSTR(string, substring)	Returns the position of the first occurrence of the substring in the given string. Returns 0, if the substring is not present in the string.	SELECT INSTR(“Informatics”, “ma”);
LTRIM(string)	Returns the given string after removing leading white space characters.	SELECT LENGTH(“ DELHI”), LENGTH(LTRIM(“ DELHI”));
RTRIM(string)	Returns the given string after removing trailing white space characters.	SELECT LENGTH(“PEN “), LENGTH(RTRIM(“PEN “));
TRIM(string)	Returns the given string after removing both leading and trailing white space characters.	SELECT LENGTH(“ MADAM “),LENGTH(TRIM(“ MADAM “));

C. Date and Time Functions:

Function	Description	Example with output
NOW()	It returns the current system date and time.	SELECT NOW();
DATE()	It returns the date part from the given date/time expression.	SELECT DATE(NOW());

MONTH(date)	It returns the month in numeric form from the date.	SELECT MONTH(NOW());
MONTHNAME(date)	It returns the month name from the specified date.	SELECT MONTHNAME(“2003-11-28”);
YEAR(date)	It returns the year from the date.	SELECT YEAR(“2003-10- 03”);
DAY(date)	It returns the day part from the date.	SELECT DAY(“2003-03- 24”);
DAYNAME(date)	It returns the name of the day from the date.	SELECT DAYNAME(“2019-07-11”);

Aggregate Functions in SQL:

Function	**Description**	**Example with output**
MAX(column)	Returns the largest value from the specified column.	SELECT MAX(Price) FROM INVENTORY;
MIN(column)	Returns the smallest value from the specified column.	SELECT MIN(Price) FROM INVENTORY;
AVG(column)	Returns the average of the values in the specified column.	SELECT AVG(Price) FROM INVENTORY;
SUM(column)	Returns the sum of the values for the specified column.	SELECT SUM(Price) FROM INVENTORY;
COUNT(*)	Returns the number of records in a table.	SELECT COUNT(*) from MANAGER;
COUNT(column)	Returns the number of values in the specified column ignoring the NULL values.	SELECT COUNT(MEMNAME) FROM MANAGER;

Group by clause in SQL: It groups the rows together that contain the same values in a specified column.

HAVING Clause in SQL is used to specify conditions on the rows with Group By clause.

Example FOR GROUP BY:

```
select rno, sum (marks)
from stdmarks
group by rno;
```

Example for having clause:

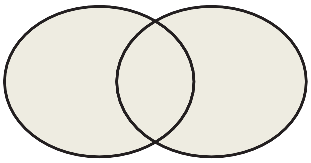
```
select rno, sum (marks)
from stdmarks
```

group by rno
having sum(marks)>100;

Join Operations on Relations: Union, Intersection, minus (set difference), Cartesian Product are the different join operations. These operations can only be applied if both the relations have the same number of attributes and corresponding attributes in both tables have the same domain.

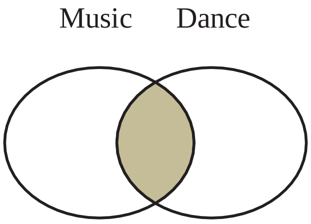
A. UNION (U) :

This operation is used to combine the selected rows of two tables at a time. If some rows are same in both the tables, then result of the Union operation will show those rows only once.

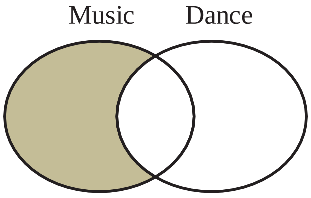


B. INTERSECT (∩)

Intersect operation is used to get the common tuples from two tables and is represented by symbol ∩. Figure shows intersection of two sets.



C. MINUS (-): This operation is used to get tuples/rows which are in the first table but not in the second table and the operation is represented by the symbol - (minus). Figure shows minus operation (also called set difference) between two sets.



D. Cartesian product (X):

Cartesian product operation combines tuples from two relations. It results in all pairs of rows from the two input relations, regardless of whether or not they have the same values on common attributes. It is denoted as ‘X’.

JOIN on two tables:

- JOIN operation combines tuples from two tables on specified conditions.
- This is unlike Cartesian product which makes all possible combinations of tuples.
- While using the JOIN clause of SQL, we specify conditions on the related attributes of two tables within the FROM clause.

SELECT * FROM UNIFORM U, COST C WHERE U.UCode = C.UCode;

Chapter 10. Computer Networks (2 MCQ + 2x1 + 3x1+5x1 (LOT) = 12 Marks)

Network: A group of two or more similar things or people interconnected with each other is called network**.**

Computer network: A computer network is an interconnection among two or more computers or computing devices.

Packets: For communication, data in a network is divided into smaller chunks called packets**.**

Media Used for computer network: Devices in a network can be connected either through wired media like cables or wireless media like air.

Node: In a communication network, each device that is a part of a network and that can receive, create, store or send data to different network routes is called a node**.**

Advantages of Computer Networking:

- Interconnection of computers allows computers to share data and resources among each other.
- Network allows sharing of resources. For example, a printer can be made available to multiple computers through a network.
- People often connect their devices through hotspot, thus forming a small personal network.
- Network allows us to exchange information simultaneously with many parties through email, websites, audio/video calls, etc.

Evolution of networking:

- In the 1960s a research project was commissioned by Advanced Research Projects Agency Network (ARPANET) in the U.S. Department of Defence to connect the academic and research institutions located at different places for scientific collaborations.
- In 1969, the first message was communicated between the University of California, Los Angeles (UCLA) and Stanford Research Institute (SRI).
- In 1971, Roy Tomlinson develops network messaging or E-mail. Symbol @ comes to mean "at".
- In 1974, the term Internet was coined, First commercial use of ARPANET, was started in the name of Telenet.
- In 1982, TCP/IP introduced as standard protocol on ARPANET.
- In 1983, Domain Name System introduced. (Ex for domain name: www.ncert.nic.in)
- In 1986, National Science Foundation brings connectivity to more people with its NSFNET program.
- In 1990, The Berners-Lee at CERN developed HTML and URL, thus giving birth to World Wide Web (www).
- In 1997, first version of Wi-Fi (802.11) standard was introduced.

Types of networks:

- PAN (Personal Area Network)
- LAN (Local Area Network)

- MAN (Metropolitan Area Network)
- WAN (Wide Area Network)

Personal Area Network (PAN):

- It is a network formed by connecting a few personal devices like computers, laptops, mobile phones, smart phones, printers etc.
- All these devices lie within an approximate range of 10 metres.
- A personal area network may be wired or wireless.
- It is owned by single user.
- For example, a mobile phone connected to the laptop through USB forms a wired PAN.

Local Area Network (LAN):

- The geographical area covered by a LAN can range from a single room, a floor, an office having one or more buildings in the same premise, laboratory, a school, college, or university campus.
- The connectivity is done by means of wires, Ethernet cables, fibre optics, or Wi-Fi.
- LAN is comparatively secure as only authentic users in the network can access other computers or shared resources.
- LANs provide the short range communication with the high speed data transfer rates.
- These types of networks can be extended up to 1 km.
- Data transfer in LAN is quite high, and usually varies from 10 Mbps (called Ethernet) to 1000 Mbps (called Gigabit Ethernet).Where Mbps stands for Megabits per second.

Metropolitan Area Network (MAN):

- Metropolitan Area Network (MAN) is an extended form of LAN which covers a larger geographical area like a city or a town.
- Data transfer rate in MAN also ranges in Mbps, but it is considerably less as compared to LAN.
- Cable TV network or cable based broadband internet services are examples of MAN.
- This kind of network can be extended up to 30-40 km.

Wide Area Network (WAN):

- Wide Area Network connects computers and other LANs and MANs,
- Which are spread across different geographical locations of a country or in different countries or continents.
- Large business, educational and government organisations connect their different branches in different locations across the world through WAN.
- The Internet is the largest WAN that connects billions of computers, smartphones and millions of LANs from different continents.

Network devices:

Modem:

- Modem stands for 'MOdulator DEModulator'.
- It refers to a device used for conversion between analog signals and digital bits.

- To transmit data from a sender to a receiver, or while browsing the internet, digital data are converted to an analog signal and the medium carries the signal to the receiver.
- There are modems connected to both the source and destination nodes.
- The modem at the sender's end acts as a modulator that converts the digital data into analog signals.
- The modem at the receiver's end acts as a demodulator that converts the analog signals into digital data for the destination node to understand.

Ethernet Card:

- Ethernet card, also known as Network Interface Card (NIC card in short) is a network adapter used to set up a wired network.
- It acts as an interface between computer and the network.
- It is a circuit board mounted on the motherboard of a computer.
- The Ethernet cable connects the computer to the network through NIC.

RJ45:

- RJ 45 or Registered Jack-45 is an eight-pin connector.
- It is used exclusively with Ethernet cables for networking.
- It is a standard networking interface that can be seen at the end of all network cables.

Repeater:

- Data are carried in the form of signals over the cable.
- These signals can travel a specified distance.
- If signals lose their strength beyond this limit and become weak. In such conditions, original signals need to be regenerated.
- A repeater is an analog device that works with signals on the cables to which it is connected.
- The weakened signal appearing on the cable is regenerated and put back on the cable by a repeater.

Hub:

- An Ethernet hub is a network device used to connect different devices through wires.
- Data arriving on any of the lines are sent out on all the others devices.
- The limitation of Hub is that if data from two devices come at the same time, they will collide.

Switch:

- When data arrives, the switch extracts the destination address from the data packet and looks it up in a table to see where to send the packet.
- It sends signals to only selected devices instead of sending to all.
- It can forward multiple packets at the same time.

Router:

- A router is a network device that can receive the data, analyses it and transmit it to other networks.

- A router connects a local area network to the internet.
- Compared to a hub or a switch, a router has advanced capabilities as it can analyse the data being carried over a network, decide/alter how it is packaged, and send it to another network of a different type.

Gateway:

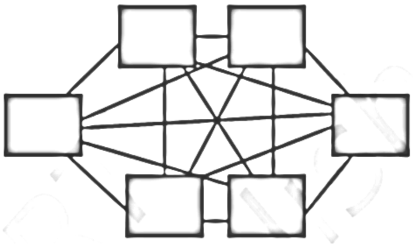
- As the term “Gateway” suggests, it is a key access point that acts as a “gate” between an organisation's network and the outside world of the Internet.
- Gateway connects two dissimilar networks. Gateway serves as the entry and exit point of a network, as all data coming in or going out of a network must first pass through the gateway in order to use routing paths.

Networking Topologies:

The arrangement of computers and other peripherals in a network is called its topology. Common network topologies are Mesh, Ring, Bus, Star and Tree.

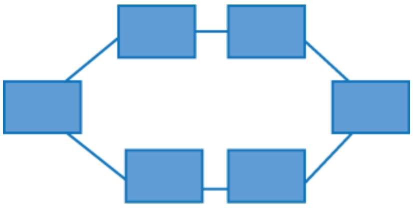
Mesh Topology:

- In this networking topology, each communicating device is connected with every other device in the network.
- This type of network can handle large amounts of traffic since multiple nodes can transmit data simultaneously.
- Such networks are more reliable in the sense that even if a node gets down, it does not cause any break in the transmission of data between other nodes.
- This topology is more secure as compared to other topologies because each cable between two nodes carries different data.



Ring Topology:

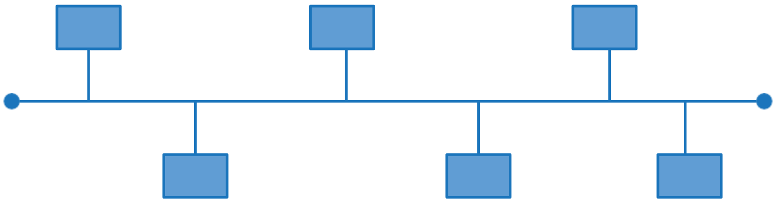
- In ring topology each node is connected to two other devices, one each on either side.
- The nodes connected with each other thus forms a ring.
- The link in a ring topology is unidirectional. Thus, data can be transmitted in one direction only (clockwise or counter clockwise).



Bus Topology:

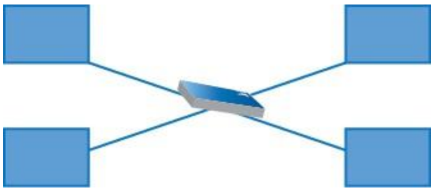
- In bus topology each communicating device connects to a transmission medium, known as bus.
- Data sent from a node are passed on to the bus and hence are transmitted to the length of the bus in both directions.
- That means, data can be received by any of the nodes connected to the bus.

- In this topology, a single backbone wire called bus is shared among the nodes, which makes it cheaper and easier to maintain.
- Both ring and bus topologies are considered to be less secure and less reliable.



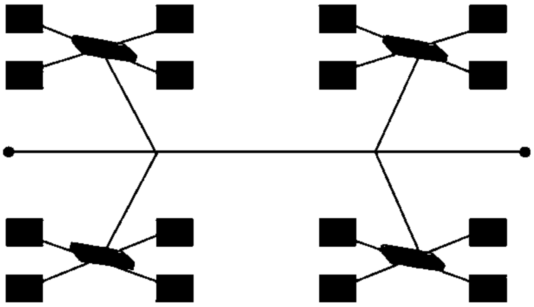
Star Topology:

- In star topology each communicating device is connected to a central node, which is a networking device like a hub or a switch.
- Star topology is considered very effective, efficient and fast as each device is directly connected with the central device.
- Disturbance in one device will not affect the rest of the network, but any failure in a central networking device may lead to the failure of complete network.
- The central node can be either a broadcasting device means data will be transmitted to all the nodes in the network.
- Or else, unicast device means the node can identify the destination and forward data to that node only.



Tree or Hybrid Topology:

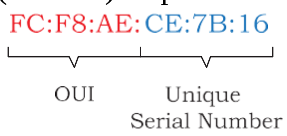
- It is a hierarchical topology, in which there are multiple branches and each branch can have one or more basic topologies like star, ring and bus.
- Such topologies are usually realised in WANs where multiple LANs are connected. Those LANs may be in the form of a ring, bus or star.
- In this type of network, data transmitted from source first reaches the centralised device and from there the data passes through every branch where each branch can have links for more nodes.



MAC Address:

- MAC stands for Media Access Control. The MAC address, also known as the physical or hardware address is a unique value associated with a network adapter called a NIC.

- The MAC address is engraved on NIC at the time of manufacturing and thus it is a permanent address and cannot be changed under any circumstances.
- The machine, on which the NIC is attached, can be physically identified on the network using its MAC address.
- Each MAC address is a 12-digit hexadecimal numbers (48 bits in length).
- First six digits (24 bits) contain the manufacturer's ID called Organisational Unique Identifier (OUI)
- Later six digits (24 bits) represent the serial number assigned to the card by the manufacturer.



IP Address:

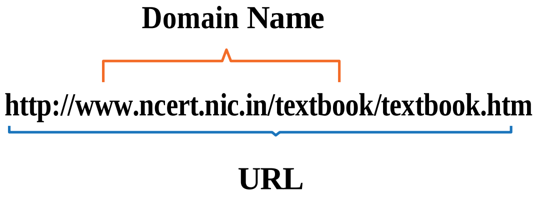
- IP address, also known as Internet Protocol address, is also a unique address that can be used to uniquely identify each node in a network.
- The IP addresses assigned to each node in a network that uses the Internet Protocol for communication.
- Thus, if we know a computer's IP address, we can communicate with that computer from anywhere in the world.
- However it is unlike MAC address, IP address can change if a node is removed from one network and connected to another network.
- Ex: 192.168.0.178

Internet, Web and the Internet of Things: The Internet is the global network of computing devices including desktop, laptop, servers, tablets, mobile phones, other handheld devices, printers, scanners, routers, switches, gateways, etc.

The World Wide Web (WWW): The World Wide Web (WWW) or web in short, is an ocean of information, stored in the form of trillions of interlinked web pages and web resources.

HTML – HyperText Markup Language: It is a language which is used to design standardized Web Pages so that the Web contents can be read and understood from any computer. Basic structure of every webpage is designed using HTML.

URI – Uniform Resource Identifier: It is a unique address or path for each resource located on the web. It is also known as Uniform Resource Locator (URL). Examples are: https://www.mhrd.gov.in,http:// www.ncert.nic.in,http://www.airindia.in, etc.



HTTP – The Hypertext Transfer Protocol: It is a set of rules which is used to retrieve linked web pages across the web. The more secure and advanced version is HTTPS.

Domain Name System:

- It works like a phonebook — we remember names instead of numbers.

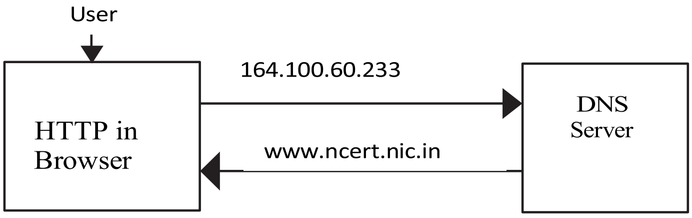
- Instead of remembering IP addresses, we assign a domain name to each IP.
- DNS connects the domain name to its correct IP address.

DNS Server:

- Instead of remembering IP addresses, we assign a domain name to each IP.
- Conversion of the domain name of each web server to its corresponding IP address is called **domain name resolution.**
- It is done through a server called DNS server.

Request of IP address corresponding to domain name:

- A DNS server maintains a database of domain names and their corresponding IP addresses.
- To understand how the domain name resolution works, we have to understand how and where the DNS servers are kept.
- The DNS servers are placed in hierarchical order. At the top level, there are 13 servers called root servers. Then below the root servers there are other DNS servers at different levels.
- A DNS server may contain the IP address corresponding to a domain or it will contain the IP address of other DNS servers, where this domain entry can be searched.



CHAPTER 11. DATA COMMUNCATION (2 mcq+2x1+(LOT)5x1=9 Marks)

Data: Data refers to raw facts it can be text, image, audio, video, and multimedia files.

Communication: Communication is an act of sending or receiving data.

Data communication: It refers to the exchange of data between two or more networked or connected devices.

Components of Data Communication:

1. **Sender:** A sender is a computer or any such device which is capable of sending data over a network. It can be a computer, mobile phone, smart watch, walkie-talkie, video recording device, etc.
2. **Receiver:** A receiver is a computer or any such device which is capable of receiving data from the network. It can be any computer, printer, laptop, mobile phone, television, etc.
3. **Message:** It is the data or information that needs to be exchanged between the sender and the receiver. Messages can be in the form of text, number, image, audio, video, multimedia, etc.
4. **Communication media:** It is also called as transmission media, the path through which the message travels between source and destination. For example, a television cable, telephone cable, ethernet cable, satellite link, microwaves, etc.
5. **Protocols:** It is a set of rules that need to be followed by the communicating parties in order to have successful and reliable data communication. Ex: Ethernet and HTTP.

Measuring Capacity of Communication Media:

- In data communication, the **transmission medium** is also called a **channel.**

- The **capacity of a channel** refers to the **maximum amount of signals or data traffic** that it can carry.

Bandwidth:

- Bandwidth of a channel is the range of frequencies available for transmission of data through that channel.
- Higher the bandwidth, higher the data transfer rate.
- Bandwidth is measured in Hertz (Hz).

Data Transfer Rate:

- Data transfer rate is the number of bits transmitted between source and destination in one second. It is also known as **bit rate.**
- It is measured in terms of bits per second (bps).

Types of Data Communication: Data communication happens in the form of signals between two or more computing devices or nodes.
Simplex communication, Half duplex communication, and Full-duplex communication.

Simplex Communication:

- It is a one way or unidirectional communication between two devices in which one device is sender and other one is receiver.
- Devices use the entire capacity of the link to transmit the data.
- Example: Keyboard input to computer, Audio from a speaker, IoT-based appliance control.

Half-duplex Communication:

- It is two way or bidirectional communication between two devices in which both the devices can send and receive data or control signals in both directions, but not at the same time.
- While one device is sending data, the other one will receive and vice-versa.
- Example: Walkie-talkies

Full-duplex Communication:

- It is two way or bidirectional communications in which both devices can send and receive data simultaneously.
- It is like a two way road where vehicles can go in both directions at the same time.
- This type of communication channel is employed to allow simultaneous communication.
- Example: Mobile phones, Landline telephones.

Switching techniques:

- Switching is a technique used to route data through a network from sender to receiver using network nodes.
- Two commonly used switching techniques are — Circuit Switching and Packet Switching.

Circuit Switching:

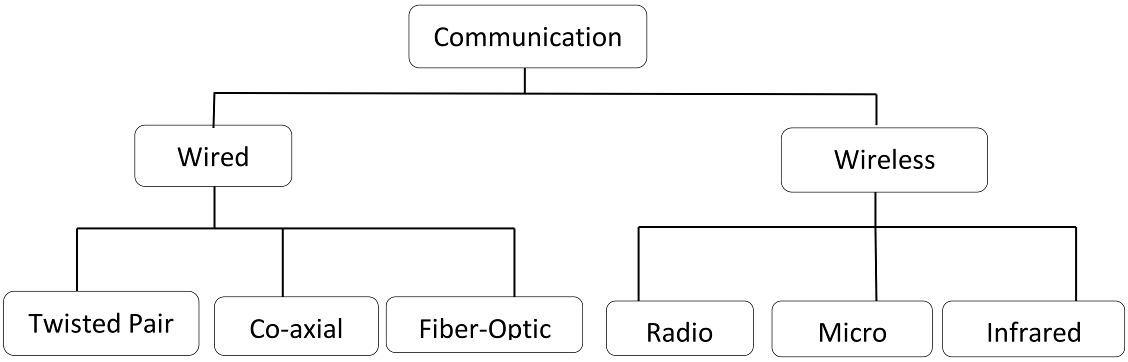
- A dedicated path is identified between the sender and the receiver.
- This path is a connected sequence of links between network nodes.
- All packets follow the same path established during the connection.
- Example: When we placed a telephone call, a physical path is established from our telephone to the receiver’s telephone.

Packet Switching:

- Each information or message to be transmitted between sender and receiver is broken down into smaller pieces, called packets. These packets are then transmitted independently through the network.
- Different packets of the same message may take different routes depending on availability.
- Each packet has two parts — a header containing the address of the destination and other information, and the main message part.
- When all the packets reach the destination, they are reassembled and the complete message is received by the receiver.

Transmission media:

- A transmission medium, which can carry signals or data between the source (transmitter) and destination (receiver).
- Example: Electric wire carries current to a fan, Air carries sound between two people talking.
- Transmission can be classified as guided or unguided.
- In guided transmission, there is a physical link made of wire/cable through which data in terms of signals are propagated between the nodes. These are usually metallic cable, fiber-optic cable, etc. They are also known as wired media.
- In unguided transmission, data travels in air in terms of electromagnetic waves using an antenna. They are also known as wireless media.



Wired Transmission Media:

- Any physical link that can carry data in the form of signals belongs to the category of wired transmission media.
- Three commonly used guided/wired media for data transmission are twisted pair, coaxial cable, and fiber optic cable.

(A) Twisted Pair Cable :

- A twisted-pair consists of two copper wires twisted like a DNA helical structure.
- Each copper wire in a twisted pair is individually insulated with a plastic cover.
- Multiple such pairs are twisted and bundled together, then enclosed within a protective outer jacket for additional safety and insulation.
- Each of the twisted pairs act as a single communication link.
- The use of twisted configuration minimizes the effect of electrical interference from similar pairs close by.
- Twisted pairs are less expensive and most commonly used in telephone lines and LANs.

- These cables are of two types: Unshielded twisted-pair (UTP) and Shielded twisted-pair (STP).

(B) Coaxial cable :

- Coaxial cable is shielded and has more bandwidth than a twisted pair.
- It has a copper wire at the core of the cable which is surrounded with insulating material.
- In a coaxial cable, the insulator is surrounded by an outer conductor.
- Backbone networks interconnect different segments of a network.
- This outer conductor is wrapped in a plastic cover.
- The key to success of coaxial cable is its shielded design that allows the cable's copper core to transmit data quickly, without interference of environmental factors.
- These types of cables are used to carry signals of higher frequencies to a longer distance.

(C) Optical Fibre :

- The optical fiber cable carries data as light. Which travels inside a thin fiber of glass.
- Optic fiber uses refraction to direct the light through the media.
- A thin transparent strand of glass at the centre is covered with a layer of less dense glass called cladding.
- This whole arrangement is covered with an outer jacket made of PVC or Teflon.

Advantages:

- ➢ This cables are used in backbone networks.
- ➢ Lightweight.
- ➢ High bandwidth → faster data transfer.
- ➢ Can carry signals over long distances.
- ➢ Immune to electromagnetic interference.
- ➢ Electromagnetic noise cannot affect the cable.

Disadvantages:

- ➢ Expensive.
- ➢ Unidirectional → needs two cables for full-duplex communication.
- ➢ Two cables are required for full duplex communication.

Wireless Transmission Media:

(A) Radio Waves:

- Radio waves of frequency 300KHz-30MHz **can travel long distance**
- Susceptible to interference
- These waves are used in AM and FM radio, television, cordless phones.

(B) Microwaves:

- Unidirectional, **can move in only one direction**.
- Cannot penetrate solid objects such as walls, hills or mountains.
- Provide very large information-carrying capacity.

(C) Infrared waves:

- Electromagnetic waves of frequency range 300GHz - 400THz.
- Very high frequency waves.
- **Cannot penetrate solid objects such as walls**.
- Used for short-distance point-to-point communication such as mobile to-mobile, mobile-to-printer, remote-control-to-TV, and Bluetooth enabled devices to other devices like mouse, keyboards etc.

Wireless Technologies:

(A) Bluetooth:

- Bluetooth is a short-range wireless technology.
- That can be used to connect mobile-phones, mouse, headphones, keyboards, computers, Bluetooth-enabled printers etc.
- Bluetooth-enabled devices have a low cost transceiver chip.
- These devices can send data within a range of 10 meters with a speed of 1 - 2 Mbps.
- Devices within range form a **personal area network (PAN)** called a **piconet.**
- Works on a **master-slave** configuration.

(B) Wireless LAN:

- A wireless **Local Area Network (LAN).**
- Defined by **IEEE 802.11**, commonly called **Wi-Fi.**
- These networks consist of communicating devices such as laptops and mobile phones, and the network device called APs (access points) which is installed in buildings or floors.

Mobile Telecommunication technologies:

1G – First Generation (1982) :

- Analog voice communication.
- Only voice calls, no data.
- Low capacity and poor security.

2G – Second Generation (1991)

- Switched from analog to digital signals.
- Improved voice quality and security (encryption).
- Supported SMS (Short Message Service) and MMS (Multimedia Messaging Service).
- Allowed more simultaneous calls.

3G – Third Generation (Commercialized in 2001)

- Supported digital voice and data services.
- Provided mobile internet access via same towers used for calls.
- Increased voice/data capacity and data speed.
- Enabled simultaneous voice and data.

4G – Fourth Generation Much faster than 3G.

- Interactive multimedia
- Voice & video
- High-speed wireless internet
- Broadband services
- Brought major improvements in user experience.

5G – Fifth Generation (Current & Emerging)

- Gbps-level data speeds
- Massive IoT and M2M (Machine to Machine) communication
- Smart cities, connected vehicles, and more
- Aims to be the backbone for future digital ecosystems.

Protocol:

- Protocol is a set of standard rules.The sender, the receiver, and all other intermediate devices need to follow.

Hypertext Transfer Protocol (HTTP):

- It is the primary protocol used to access the World Wide Web.
- Tim Berners-Lee led the development of HTTP at CERN in 1989 in collaboration with Internet Engineering Task Force (IETF) and the World Wide Web Consortium (W3C).
- HTTP is a request-response (also called client server) protocol that runs over TCP.
- The common use of HTTP is between a web browser (client) and a web server (server).
- HTTP facilitates access of hypertext from the World Wide Web by defining how information are formatted and transmitted, and how the Web servers and browsers should respond to various commands.

File Transfer Protocol (FTP):

- Protocol used for transferring files from one machine to another.
- FTP also works on a client-server model.
- When a user requests for a file transfer with another system, FTP sets up a connection between the two nodes for accessing the file.
- The user can authenticate using user ID and password.

Point to Point Protocol (PPP):

- **PPP is a communication protocol which establishes a dedicated and direct connection between two communicating devices.**
- This protocol defines how two devices will authenticate each other and establish a direct link between them to exchange data.
- PPP is commonly used in:
  - ➢ Direct communication between two routers.
  - ➢ Internet users connecting their home computers to an ISP server via a modem.

Simple Mail Transfer Protocol (SMTP):

- It uses information written on the message header (like an envelope on a letter sent by post), and is not concerned with the content of the email message.
- Each email header contains email addresses of recipients.
- The email containing header and body are entered into a queue of outgoing mails.
- The SMTP sender program takes mails from the outgoing queue and transmits them to the destination.
- When the SMTP sender successfully delivers a particular mail to one or more destinations, it removes the corresponding receiver’s email address from the mail’s destination list.
- When that mail is delivered to all the recipients, it is removed from the outgoing queue.
- The SMTP receiver program accepts each mail that has arrived and places it in the appropriate user mailbox.

Transmission Control Protocol (TCP)/ Internet Protocol (IP):

- It is a set of standardized rules that uses a client-server model of communication in which a user or machine (a client) requests a service by a server in the network.
- The IP protocol ensures that each computer or node connected to the Internet is assigned an IP address, which is used to identify each node independently.
- TCP ensures that the message or data is broken into smaller chunks, called IP packets.

- Each of these packets is routed (transmitted) through the Internet, along a path from one router to the next, until it reaches the specified destination.
- TCP guarantees the delivery of packets on the designated IP address. It is also responsible for ordering the packets so that they are delivered in sequence.
- When all the packets finally reach the destination machine, they are reassembled into the original message at the receiver's end.

Chapter12. Security Aspects(1 mcq+(LOT)5x1=6 Marks)

Malware

- Malware is a short term used for **MAL**icious soft**WARE**.
- It is any software developed with an intention to damage hardware devices, steal data, or cause any other trouble to the user.
- Viruses, Worms, Ransomware, Trojans, and Spyware are some of the kinds of malware.

1. Virus:

- The term computer virus was coined by Fred Cohen in 1985.
- A virus is a piece of software code created to perform malicious activities and hamper resources of a computer system like CPU time, memory, personal files, or sensitive information. Ex:CryptoLocker, MyDoom, Sasser and Netsky,Slammer, Stuxnet, etc.

2. **Worms:**

- The Worm is also a malware that incurs unexpected or damaging behavior on an infected computer system.
- The major difference between a worm and a virus is that unlike a virus, a worm does not need a host program or software to insert its code into.
- Worms are standalone programs that replicates on its own and can spread to other computers through the network.
- Ex: Storm Worm, Sobig, MSBlast, Code Red, Nimda, Morris Worm, etc

3. Ransomware:

- It is a type of malware that targets user data.
- It either blocks the user from accessing their own data or threatens to publish the personal data online and demands ransom payment against the same.

4. Trojan:

- Trojan is a malware, that looks like legitimate software and once it tricks a user into installing it, it acts pretty much like a virus or worm.
- However, a Trojan does not self-replicate or infect other files, it spreads through user interaction such as opening an email attachment or downloading and executing a file from the Internet.

5. Spyware

- It is a type of malware that spies on a person or an organization by gathering information about them, without the knowledge of the user.

- It records and sends the collected information to an external entity without consent or knowledge of the user.
- Spyware usually tracks internet usage data and sells them to advertisers.
- They can also be used to track and capture credit card or bank account information, login and password information or user’s personal identity.

6. Adware

- An Adware is a malware that is created to generate revenue for its developer.
- An adware displays online advertisements using pop-ups, web pages, or installation screens.
- Adware is usually annoying, but harmless.

Keyloggers

- A keylogger can either be malware or hardware.
- The main purpose of this malware is to record the keys pressed by a user on the keyboard.
- A keylogger makes logs of daily keyboard usage and may send it to an external entity as well.
- In this way, very sensitive and personal information like passwords, emails, private conversations, etc. can be revealed to an external entity without the knowledge of the user.

Modes of Malware distribution

- Downloaded from the Internet: Most of the time, malware is unintentionally downloaded into the hard drive of a computer by the user.
- Spam Email: We often receive an unsolicited email with embedded hyperlinks or attachment files. These links or attached files can be malware.
- Removable Storage Devices: Often, the replicating malware targets the removable storage media like pen drives, SSD cards, music players, mobile phones, etc. and infect them with malware that gets transferred to other systems that they are plugged into.
- Network Propagation: Some malware like Worms have the ability to propagate from one computer to another through a network connection.

Combating Malware: means protecting a computer or network from harmful software and removing it if it infects the system.

- Frequent pop-up windows prompting you to visit some website and/or download some software;
- Changes to the default homepage of your web browser;
- Mass emails being sent from your email account;
- Unusually slow computer with frequent crashes;
- Unknown programs startup as you turn on your computer;
- Programs opening and closing automatically;
- Sudden lack of storage space, random messages, sounds, or music start to appear;
- Programs or files appear or disappear without your knowledge.

Some of the preventive measures against the malware:

- Using antivirus, anti-malware, and other related software and updating them on a regular basis.
- Configure your browser security settings
- Always check for a lock button in the address bar while making payments.
- Never use pirated on unlicensed software. Instead go for Free and Open Source Software
- Applying software updates and patches released by its manufacturers.
- Taking a regular backup of important data.
- Enforcing firewall protection in the network.
- Avoid entering sensitive or personal information on unknown or public computers.
- Avoid clicking on links or downloading attachments from unsolicited emails.
- Never share your online account or banking password/pins with anyone.

Antivirus

- Antivirus is a software, also known as anti-malware.
- Antivirus software was developed to detect and remove viruses only and hence the name antivirus.

Methods of Malware Identification used by Antivirus:

- **Signature-based detection:** In this method, an antivirus works with the help of a signature database known as "Virus Definition File (VDF)". This file consists of virus signatures and is updated continuously on a real-time basis. This makes the regular update of the antivirus software a must.

- **Sandbox detection :**In this method, a new application or file is executed in a virtual environment (sandbox) and its behavioral fingerprint is observed for a possible malware. Depending on its behavior, the antivirus engine determines if it is a potential threat or not and proceeds accordingly.

- **Data mining techniques:** This method employs various data mining and machine learning techniques to classify the behavior of a file as either benign or malicious.

- **Heuristics:** Often, a malware infection follows a certain pattern. Here, the source code of a suspected program is compared to viruses that are already known and are in the heuristic database. If the majority of the source code matches with any code in the heuristic database, the code is flagged as a possible threat.

- **Real-time protection:**In this technique, the anti-malware software keeps running in the background and observes the behavior of an application or file for any suspicious activity while it is being executed i.e. when it resides in the main memory of the computer system.

Spam

- Spam is a broad term and applies to various digital platforms like messaging, forums, chatting, emailing, advertisement, etc.
- However, the widely recognized form is email spam.

- Depending on their requirements, organisations or individuals buy or create a mailing list (list of email addresses) and repeatedly send advertisement links and invitation emails to a large number of users.
- This creates unnecessary junk in the inbox of the receiver's email and often tricks a user into buying something or downloading a paid software or malware.

HTTP vs. HTTPS

- Both the HTTP (Hyper Text Transfer Protocol) and its variant HTTPS (Hyper Text Transfer Protocol Secure) **are a set of rules** (protocol) **that govern how data can be transmitted over the WWW** (World Wide Web).
- HTTP sends information over the network as it is.
- HTTPS encrypts the data before transmission. At the receiver end, it decrypts to recover the original data.

Firewall

- Computer firewall is a network security system designed to protect a trusted private network from unauthorized access from untrusted outside network.
- Firewall can be implemented in software, hardware or both.
- A firewall acts as a network filter and based on the predefined security rules, it continuously monitors and controls the incoming and outgoing traffic.
- As an example, a rule can be set in the firewall of a school LAN, that a student cannot access data from the finance server, while the school accountant can access the finance server.

Types of Firewall

- **Network Firewall**: If the firewall is placed between two or more networks and monitors the network traffic between different networks, it is termed as Network Firewall.
- **Host-based Firewall**: If the firewall is placed on a computer and monitors the network traffic to and from that computer, it is called a host-based firewall.

Cookies

- A computer cookie is a small file or data packet, which is stored by a website on the client's computer.
- A cookie is edited only by the website that created it, the client's computer acts as a host to store the cookie.
- Cookies are used by the websites to store browsing information of the user.

Threats due to Cookies

- Usually, cookies are used for enhancing the user's browsing experience and do not infect your computer with malware.
- However, some malware might disguise as cookies e.g. "supercookies".
- There is another type of cookie known as "Zombie cookie" that gets recreated after being deleted.

- As a common example, if you search for a particular item using your search engine, a third-party cookie will display advertisements showing similar items on other websites that you visit later.
- So, one should be careful while granting permission to any websites to create and store cookies on the user computer.

Hackers and Crackers

- Hackers and crackers are people having a thorough knowledge of the computer systems, system software (operating system), computer networks, and programming. They use this knowledge to find loopholes and vulnerabilities in computer systems or computer networks and gain access to unauthorized information.
- Hacker is a person that is skilled enough to hack or take control of a computer system.

Types of Hackers:

a) White Hats: Ethical Hacker

- If a hacker uses his/her knowledge to find and help in fixing the security flaws in the system, its termed as White Hat hacker.
- These are the hackers with good intentions. They are actually security experts.

b) Black Hats: Crackers

If hackers use their knowledge unethically to break the law and disrupt security by exploiting the flaws and loopholes in a system, then they are called black hat hackers.

c) Grey Hats

The grey hats take system security as a challenge and just hack systems for the fun of it.

Network Security Threats

Denial of Service

- Denial of Service (DoS) is a scenario, wherein an attacker (Hacker) limits or stops an authorized user to access a service, device, or any such resource by overloading that resource with illegitimate requests.
- If a DoS attack makes a server crash, the server or resource can be restarted to recover from the attack. However, a flooding attack is difficult to recover from, as there can be some genuine legitimate requests in it as well.
- A variant of DoS, known as Distributed Denial of Service (DDoS) is an attack, where the flooded requests come from compromised computer (Zombies) systems distributed across the globe or over a very large area.
- The attacker installs malicious software known as Bot on the Zombie machines, which gives it control over these machines.
- Depending upon the requirement and availability, the attacker activates a network of these Zombie computers known as Bot-Net to carry out the DDoS attack.

Intrusion Problems:

A. Asymmetric Routing

The attacker tends to avoid detection by sending the intrusion packets through multiple paths, thereby bypassing the network intrusion sensors.

B. Buffer Overflow Attacks

In this attack, the attacker overwrites certain memory areas of the computers within the network with code that will be executed later when the buffer overflow occurs.

C. Traffic Flooding

It involves flooding the network intrusion detection system with message packets. This huge load leaves the network detection system incapable of monitoring the packets adequately.

Snooping

- Snooping means secretly listening to a conversation.
- In the context of networking, it refers to the process of secret capture and analysis of network traffic.
- In this attack, the hacker taps or listens to a channel of communication by picking all of the traffic passing through it.
- Once the network packets are analysed by the snooping device or software, it reproduces the exact traffic packets and places them back in the channel, as if nothing has happened.
- However, snooping is not always an attack, at times it is also used by network administrators for troubleshooting various network issues.
- Snooping is also known as Sniffing.

Eavesdropping

- The term eavesdropping has been derived from the literal practice of secretly listening to the conversations of people by standing under the eaves of a house.
- Unlike snooping, where the network traffic can be stored for later analysis, eavesdropping is an unauthorised real-time interception or monitoring of private communication between two entities over a network.
- Also, the targets are usually the private communication channels like phone calls, instant messages, video conference, fax transmission, etc.
- In older days, eavesdropping was performed on the conventional telephone line and was known as wiretapping.
- Eavesdropping is different from Snooping. While the eavesdropping happens in real time, but snooping does not.
- As an example, in eavesdropping, imagine someone listening to your private conversation with the help of a hidden microphone in your room or by physically standing near the window of your room.
- However, in snooping, that person may make a copy of a letter that is addressed to your friend and keep the copy with himself and send the original letter to the intended address.