

# केन्द्रीय विद्यालय संगठन, क्षेत्रीय कार्यालय, राँची

Kendriya Vidyalaya Sagathan, Regional Office, Ranchi



**CLASS XII**

**Computer Science (083)**

**STUDY MATERIAL**

**Session: 2023-2024**

## OUR PATRON

Sh. D. P. Patel  
Deputy Commissioner  
Kendriya Vidyalaya Sangathan, Ranchi Region  
Chief Patron

Smt. Sujata Mishra Assistant  
Commissioner  
Kendriya Vidyalaya Sangathan, Ranchi Region  
Patron

Sh. R C Gond  
Principal  
Kendriya Vidyalaya Patratu  
Convener

## CONTENT TEAM

<u>NAME OF TEACHER</u>	<u>DESIGNATION</u>	<u>NAME OF KV</u>
Mr. K Chakrovortty	PGT CS	K V GOMOH
Mr. Amod Kumar	PGT CS	K V PATRATU
Mr. Ravi Prakash	PGT CS	K V DIPATOLI
Mr. Sanjay Kumar Sinha	PGT CS	K V CCL
Mr. Lalit Kumar Jha	PGT CS	K V Bokaro No 1
Mr. R N P Sinha	PGT CS	K V HINOO 2ND SHIFT
Mr. Rakesh Kumar Pathak	PGT CS	K V BOKARO THERMAL
Mr. Aftab Alam Shahnwaz	PGT CS	K V RAMGARH
Mr. Pravin Kumar Singh	PGT CS	K V DHANDAB NO 1
Mr. Mantosh Kumar	PGT CS	K V DEEPATOLI
Mr. Murtaza Ali	PGT CS	K V GODDA
Mr. Anand Prasad	PGT CS	K V JAMTARA

## **INDEX**

SL NO.	TOPIC	PAGE
<b>01</b>	Revision of Python topics	1
<b>02</b>	Functions & Exception Handling	21
<b>03</b>	Introduction to files & Text Files	32
<b>04</b>	Binary Files	44
<b>05</b>	CSV files	57
<b>06</b>	Data Structures (Stack)	64
<b>07</b>	Computer Network	74
<b>08</b>	Database Management System	94
<b>09</b>	Interface of python with SQL database:	115

## Revision Tour -1

**Data Types:** Data Type specifies which type of value a variable can store. `type()` function is used to determine a variable's type in Python

Data Types In Python

1. Number
2. String
3. Boolean
4. List
5. Tuple
6. Set
7. Dictionary

**Python tokens :**

(1) keyword :

Keywords are reserved words. Each keyword has a specific meaning to the Python interpreter, and we can use a keyword in our program only for the purpose for which it has been defined. As Python is case sensitive, keywords must be written exactly.

(2) Identifier : Identifiers are names used to identify a variable, function, or other entities in a program. The rules for naming an identifier in Python are as follows:

- The name should begin with an uppercase or a lowercase alphabet or an underscore sign (`_`). This may be followed by any combination of characters `a-z`, `A-Z`, `0-9` or underscore (`_`). Thus, an identifier cannot start with a digit.

- It can be of any length. (However, it is preferred to keep it short and meaningful).

It should not be a keyword or reserved word

- We cannot use special symbols like `!`, `@`, `#`, `$`, `%`, etc., in identifiers.

(3) Variables: A variable in a program is uniquely identified by a name (identifier). Variable in Python refers to an object — an item or element that is stored in the memory. Comments: Comments are used to add a remark or a note in the source code. Comments are not executed by interpreter. a comment starts with `#` (hash sign). Everything following the `#` till the end of that line is treated as a comment and the interpreter simply ignores it while executing the statement.

Mutable and immutable data types : Variables whose values can be changed after they are created and assigned are called mutable. Variables whose values cannot be changed after they are created and assigned are called immutable.



(4) Operators: An operator is used to perform specific mathematical or logical operation on values. The values that the operators work on are called operands.

Arithmetic operators :four basic arithmetic operations as well as modular division, floor division and exponentiation. `(+, -, *, /)` and `(%, //, **)`

Relational operators : Relational operator compares the values of the operands on its either side and determines the relationship among them. `==, !=, >, <, <=, >=`

Logical operators : There are three logical operators supported by Python. These operators (`and, or, not`) are to be written in lower case only. The logical operator

evaluates to either True or False based on the logical operands on either side. and , or , not

Assignment operator : Assignment operator assigns or changes the value of the variable on its left. a=1+2 Augmented assignment operators : += , -= , /= \*= , //= %= , \*\*=

Identity operators : is, is no

Membership operators : in, not in

### **Type Conversion:**

The process of converting the value of one data type (integer, string, float, etc.) to another data type is called type conversion. Python has two types of type conversion.

- Implicit Type Conversion /automatic type conversion
- Explicit Type Conversion

### **CONTROL STATEMENTS**

Control statements are used to control the flow of execution depending upon the specified condition/logic.

There are three types of control statements:

1. Decision Making Statements (if, elif, else)
2. Iteration Statements (while and for Loops)
3. Jump Statements (break, continue, pass)

### **Questions and Answers**

#### **1 Mark questions**

Q1. Which of the following is not considered a valid identifier in Python:

- (i)three3      (ii)\_main      (iii)hello\_kv1      (iv)2 thousand

Q2. Which of the following is the mutable data type in python:

- (i)int      (ii) string      (iii)tuple      (iv)list

Q3. Which of the following statement converts a tuple into a list in Python:

- (i) len(string)      (ii)list(tuple)      (iii)tup(list)      (iv)dict(string)

Q4. Name of the process of arranging array elements in a specified order is termed as

- i)indexing      ii)slicing      iii)sorting      iv) traversing

Q5. What type of value is returned by input() function by default?

- i)int      ii)float      iii)string      iv)list

Q6. Which of the following operators cannot be used with string

- (i) +      (ii) \*      (iii) -      (iv) All of these

Q7. If L = [0.5 \* x for x in range(0,4)] , L is

- i)[0,1,2,3]      ii)[0,1,2,3,4]      iii) [0.0,0.5,1.0,1.5]      iv) [0.0,0.5,1.0,1.5,2.0]

Q8. Write the output of the following python code:

```
x = 123
for i in x:
    print(i)
```

- i)1 2 3      ii) 123      iii)infinite loop      iv) error

Q9. write the output of following code

```
A = 10/2
B = 10/3
print(A,B)
```

- i)5,3.3      ii) 5.0 , 3.3      iii) 5.0 , 3      iv) 5,4

Q10. Name the built-in mathematical function / method that is used to return square root of a number.

- i)SQRT()      ii)sqrt()      iii) sqt()      iv) sqte()

## 2 MARK QUESTIONS

Q1. Find the following python expressions:

a)  $(3-10^{**2}+99/11)$

b)  $\text{not } 12 > 6 \text{ and } 7 < 17 \text{ or not } 12 < 4$

c)  $2^{**3}^{**2}$

d)  $7 // 5 + 8 * 2 / 4 - 3$

Q2. i) Convert the following for loop into while loop

```
for i in range(10,20,5):
```

```
    print(i)
```

ii) Evaluate:- not false and true or false and true

Q3. What are advantages of using local and global variables ?

Q4. Remove the errors from the following code Rewrite the code by underlining the errors .

```
x = int(("enter the value"))
```

```
for i in range [0,11]:
```

```
    if x = y
```

```
        print x+y
```

```
    else:
```

```
        print x-y
```

Q5. Rewrite the following code in python after removing all syntax errors. Underline each correction done in the code:

```
def func(x):
```

```
    for i in (0,x):
```

```
        if i%2 =0:
```

```
            p=p+1
```

```
        else if i%5 = =0
```

```
            q=q+2
```

```
        else:
```

```
            r=r+i
```

```
    print(p,q,r)
```

```
    func(15)
```

Q6. Write the output of the following code:-

```
String1="Coronavirus Disease"
```

```
print(String1.lstrip("Covid"))
```

```
print(String1.rstrip("sea"))
```

Q7. Write the ouput of the following code:-

```
Text = "gmail@com"
```

```
L=len(Text)
```

```
Ntext=""
```

```
for i in range(0,L):
```

```
    if Text[i].isupper():
```

```
        Ntext=Ntext+Text[i].lower()
```

```
    elif Text[i].isalpha():
```

```
        Ntext= Ntext+Text[i].upper()
```

```
    else:
```

```
        Ntext=Ntext+'bb'
```

```
print(Ntext)
```

Q8.  $L = [“abc”, [6,7,8], 3, “mouse”]$

Perform following operations on the above list L.

i)  $L[3:]$

ii)  $L[: : 2]$

iii)  $L[1:2]$

iv)  $L[1][1]$

Q9. Write the output of the following:

```
word = 'green vegetables'
```

```
print(word.find('g',2))
```

```
print(word.find('veg',2))
```

```
print(word.find('tab',4,15))
```

### 3 MARK QUESTIONS

Q1. Write the python program to print the index of the character in a string.

Example of string : "pythonProgram"

Expected output:

Current character p position at 0

Current character y position at 1

Current character t position at 2

Q2. Find and write the output of the following python code:

```
string1 = "Augmented Reality"
```

(i) print(string1[0:3])

(ii) print(string1[3:6])

(iii) print(string1[:7])

(iv) print(string1[-10:-3])

(v) print(string1[-7: :3]\*3)

(vi) print(string1[1:12:2])

Q3. Find the output of the give program :

```
x = "abcdef"
```

```
j = "a"
```

```
for i in x:
```

```
    print(j, end = " ")
```

Q4. Find output generated by the following code:

```
i = 3
```

```
while i >= 0:
```

```
    j = 1
```

```
    while j <= i:
```

```
        print(j,end = ' ')
```

```
        j = j + 1
```

```
    print()
```

```
    i = i - 1
```

Q5. Find output generated by the following code:

```
i = 1
```

```
y = 65
```

```
while i<=5:
```

```
    j = i
```

```
    while j<=I:
```

```
        print(chr(y),end=' ')
```

```
        j= j+1
```

```
        y = y+1
```

```
    print()
```

```
    i=i+1
```

4

### MARK QUESTIONS

Q1.Differentiate between break and continue statement used in python.

Q2.What is comment in python ? Explain its significance.

Q3.Explain the types of errors occurring in python programming language.

5

### MARK QUESTIONS

Q1.Differentiate between type conversion and type casting in python with examples.

Q2.Explain mutable and immutable objects in python with examples.

Q3. What is the use of else statement in for loop and in while loop ? Explain.

### ANSWERS

#### ANSWER OF 1 MARK QUESTIONS

1) (iv)

2) (iv)

3) (ii)

- 4) (iii)
- 5) (iii)
- 6) (iii)
- 7) (iii)
- 8) (iv)
- 9) (iii)
- 10) (ii)

### **ANSWER OF 2 MARK QUESTIONS**

1) a) -88.0 b) True c)512 d)2.0

2) (i) i=10

```
while(i<20):
    print(i)
    i+=5
```

(i) true

3. Advantages of Local Variable

- o The same name of a local variable can be used in different functions as it is only recognized by the function in which it is declared.
- o Local variables use memory only for the limited time when the function is executed; after that same memory location can be reused.

Advantages of Global Variable

- o Global variables can be accessed by all the functions present in the program.
- o Only a single declaration is required.
- o Very useful if all the functions are accessing the same data.

4.

```
x = int(input("enter the value"))
for y in range(0,11):
    if x == y:
        print(x+y)
    else:
        print(x - y)
```

5. def func(x): Error 1

for i in range(0,x): Error 2

if i%2 ==0: Error 3

p=p+1

elif i%5 ==0 Error 4

q=q+2

else:

r=r+i

print(p,q,r)

func(15)else: Error 4

print(c+10)

6. ronavirus Disease

Coronavirus Di

7. GMAILbbCOM

8. I) ['mouse'] ii) ['abc',3] iii)[ [ 6,7,8] ] iv) 7

9. 8

6

10

### **ANSWER OF 3 MARK QUESTIONS**

1) string1 = input("enter string")

for i in range(len(string1)):

print("current character",string1[i],"position at",i)



- 2) (i) Aug (ii) men (iii) Augment (iv)ed Real (v)RlyRlyRly (vi) umne e  
 3) aaaaaa  
 4) 1 2 3  
     1 2  
     1  
 5) A  
     BC  
     DEF  
     GHIJ  
     KLMNO

### ANSWER OF 4 MARK QUESTIONS

- 1) The break statement terminates the current loop , i.e the loop in which it appears, and resumes execution at the next statement immediately after the end of that loop.if break statement is inside a nested loop(loop inside another loop), break will terminate the innermost loop.

When a continue statement is encountered, the control jumps to the beginning of the loop for next iteration, thus skipping the execution of statements inside the body of loop for the current iteration. As usual, the loop condition is checked to see if the loop should continue further or terminate. If the condition of the loop is entered again, else the control is transferred to the statement immediately following the loop.

- 2) Comments in Python are identified with a **hash** symbol, #, and extend to the end of the line. Hash characters in a string are not considered comments, however. There are three ways to write a comment - as a separate line, beside the corresponding statement of code, or as a multi-line comment block.

here are multiple uses of writing comments in Python. Some significant uses include:

- Increasing readability
- Explaining the code to others
- Understanding the code easily after a long-term
- Including resources
- Re-using the existing code

3. There are three types of Python errors.

1. Syntax errors

Syntax errors are the most basic type of error. They arise when the Python parser is unable to understand a line of code. Syntax errors are almost always fatal, i.e. there is almost never a way to successfully execute a piece of code containing syntax errors.

2.Logical errors

These are the most difficult type of error to find, because they will give unpredictable results and may crash your program. A lot of different things can happen if you have a logic error.

3.Run time errors

Run time errors arise when the python knows what to do with a piece of code but is unable to perform the action.Since Python is an interpreted language, these errors will not occur until the flow of control in your program reaches the line with the problem. Common example of runtime errors are using an undefined variable or mistyped the variable name.

### ANSWER OF 5 MARK QUESTIONS

1. Type Conversion

In type conversion, the python interpreter automatically converts one data type to another. Since Python handles the implicit data type conversion, the programmer does not have to convert the data type into another type explicitly.

The data type to which the conversion happens is called the destination data type, and the data type from which the conversion happens is called the source data type.

In type conversion, the destination data of a smaller size is converted to the source data type of larger size. This avoids the loss of data and makes the conversion safe to use.

```
x = 20
y = 25.5
Z = x + y
```

Here value in z, int type is converted to float type

### Type Casting

In type casting, the programmer has to change the data type as per their requirement manually. In this, the programmer explicitly converts the data type using predefined functions like int(), float(), str(), etc. There is a chance of data loss in this case if a particular data type is converted to another data type of a smaller size.

```
x = 25
float(x)
```

It converts into float type

2. Mutable in Python can be defined as the object that can change or be regarded as something changeable in nature. Mutable means the ability to modify or edit a value.

Mutable objects in Python enable the programmers to have objects that can change their values. They generally are utilized to store a collection of data. It can be regarded as something that has mutated, and the internal state applicable within an object has changed. Immutable objects in Python can be defined as objects that do not change their values and attributes over time.

These objects become permanent once created and initialized, and they form a critical part of data structures used in Python.

Python is used in numbers, tuples, strings, frozen sets, and user-defined classes with some exceptions. They cannot change, and their values and it remains permanent once they are initialized and hence called immutable.

3. Else with loop is used with both while and for loop. The else block is executed at the end of loop means when the given loop condition is false then the else block is executed.

```
i = 0
while i < 5:
    i += 1
    print("i =", i)
else:
    print("else block is executed")
```

#### Explanation

- declare i=0
- we know then while loop is active until the given condition is true. and we check i<5 it's true till the value of i is 4.
- i+=1 increment of i because we don't want to execute the while loop infinite times.
- print the value of i
- else block execute when the value of i is 5.

```
l = [1, 2, 3, 4, 5]
for a in l:
    print(a)
else:
    print("else block is executed")
```

#### Explanation

- declare a list l=[1,2,3,4,5]
- for loop print a.
- else block is execute when the for loop is read last element of list.

## STRINGS

A sequence of characters is called a string. Strings are used by programming languages to manipulate text such as words and sentences.

Strings literal in Python are enclosed by double quotes or single quotes. String literals can span multiple lines, to write these strings triple quotes are used.

```
>>> a = ''' Python
Programming
Language'''
Empty string can also be created in
Python .
>>> str = ''
```

### Accessing Values in Strings

Each individual character in a string can be assessed using a technique called indexing .

Python allows both positive and negative indexing.

```
S = "Python Language"
```

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
P	y	t	h	o	n		L	a	n	g	u	a	g	e
-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

```
>>> S[7]          >>>S[-10]
L                n
```

### Deleting a String

As we know, strings are immutable, so we cannot delete or remove the characters from the string but we can delete entire string using del keyword.

```
>>> str1 = " WELCOME "
>>> del str1
>>> print ( str1 )
NameError : name ' str1 ' is not defined .
```

### String Slicing

To access some part of a string or substring, we use a method called slicing.

Syntax: string\_name[start : stop]

```
>>> str1 = " Python Program "
>>> print ( str1[ 3: 8] )          >>> print ( str1 [ : -4 ] )          >>> print ( str1 [ 5 : ] )
hon P                            Python Pro                            n Program
```

Strings also provide slice steps which used to extract characters from string that are not consecutive. Syntax string\_name [ start : stop : step ]

```
>>> print ( str1 [ 2 : 12 : 3 ] )
tnrr
```

We can also print all characters of string in reverse order using [::-1]

```
>>> print ( str1 [ :: - 1 ] )
margorP nohtyP
```

### Traversing a String

1. Using 'for' loop: for loop can iterate over the elements of a sequence or string . It is used when you want to traverse all characters of a string. eg.

```
>>> sub = " GOOD "
>>> for i in sub:
    print ( i )
Output:
G
O
O
D
```

2. Using 'for' loop with range

```
>>> sub = " COMPUTER "
>>> for i in range ( len ( sub ) ) :
    print ( sub [ i ], '-', end = ' ' )
Output: Output C - O - M - P - U - T - E - R
```

### String Operations

#### Concatenation

To concatenate means to join. Python allows us to join two strings using the concatenation operator plus which is denoted by symbol +.

```
>>> str1 = 'Hello' #First string
>>> str2 = 'World!' #Second string
>>> str1 + str2 #Concatenated strings 'HelloWorld!'
```

### String Replication Operator ( \* )

Python allows us to repeat the given string using repetition operator which is denoted by symbol (\*).

```
>>> a = 2 * " Hello "
>>> print ( a )
HelloHello
```

### Comparison Operators

Python string comparison can be performed using comparison operators ( == , > , < , , <= , >= ). These operators are as follows

```
>>> a = 'python ' < ' program '
>>> print ( a )
False
>>> a = ' Python '
>>> b = ' PYTHON "
>>> a > b
True
```

**Membership Operators** are used to find out whether a value is a member of a string or not .

#### (i) in Operator:

```
>>> a = "Python Programming Language"
>>> "Programming" in a
True
```

#### (ii) not in Operator:

```
>>> a = "Python Programming Language"
>>> "Java" not in a
True
```

### BUILT IN STRING METHODS

Method	Description
len()	Returns length of the given string
title()	Returns the string with first letter of every word in the string in uppercase and rest in lowercase
lower()	Returns the string with all uppercase letters converted to lowercase
upper()	Returns the string with all lowercase letters converted to uppercase
count(str, start, end)	Returns number of times substring str occurs in the given string.
find(str,start, end)	Returns the first occurrence of index of substring stroccurring in the given string. If the substring is not present in the given string, then the function returns -1
index(str, start, end)	Same as find() but raises an exception if the substring is not present in the given string
endswith()	Returns True if the given string ends with the supplied substring otherwise returns False
startswith()	Returns True if the given string starts with the supplied substring otherwise returns False
isalnum()	Returns True if characters of the given string are either alphabets or numeric. If whitespace or special symbols are part of the given string or the string is empty it returns False
islower()	Returns True if the string is non-empty and has all lowercase alphabets, or has at least one character as lowercase alphabet and rest are non-alphabet characters

isupper()	Returns True if the string is non-empty and has all uppercase alphabets, or has at least one character as uppercase character and rest are non-alphabet characters
isspace()	Returns True if the string is non-empty and all characters are white spaces (blank, tab, newline, carriage return)
istitle()	Returns True if the string is non-empty and title case, i.e., the first letter of every word in the string in uppercase and rest in lowercase
lstrip()	Returns the string after removing the spaces only on the left of the string
rstrip()	Returns the string after removing the spaces only on the right of the string
strip()	Returns the string after removing the spaces both on the left and the right of the string
replace (oldstr, newstr)	Replaces all occurrences of old string with the new string
join()	Returns a string in which the characters in the string have been joined by a separator
partition ( )	Partitions the given string at the first occurrence of the substring (separator) and returns the string partitioned into three parts. 1. Substring before the separator 2. Separator 3. Substring after the separator If the separator is not found in the string, it returns the whole string itself and two empty strings
split()	Returns a list of words delimited by the specified substring. If no delimiter is given then words are separated by space.

## LIST

List is an ordered sequence, which is used to store multiple data at the same time. List contains a sequence of heterogeneous elements. Each element of a list is assigned a number to its position or index. The first index is 0 (zero), the second index is 1, the third index is 2 and so on.

### **Creating a List In Python,**

```
a = [ 34 , 76 , 11,98 ]
```

```
b = [ ' s ' , 3 , 6 , ' t ' ]
```

```
d = [ ]
```

**Creating List From an Existing Sequence:** list ( ) method is used to create list from an existing sequence . Syntax: new\_list\_name = list ( sequence / string )

You can also create an empty list . eg . a = list ( ) .

### **Similarity between List and String**

- len ( ) function is used to return the number of items in both list and string .
- Membership operators as in and not in are same in list as well as string .
- Concatenation and replication operations are also same done in list and string.

### **Difference between String and List**

Strings are immutable which means the values provided to them will not change in the program. Lists are mutable which means the values of list can be changed at any time.

### **Accessing Lists**

To access the list's elements, index number is used.

```
S = [12,4,66,7,8,97,"computer",5.5,]
```

```
>>> S[5]
```

```
97
```

```
>>>S[-2]
```

```
computer
```

0	1	2	3	4	5	6	7
12	4	66	7	8	97	computer	5.5
-8	-7	-6	-5	-4	-3	-2	-1

## Traversing a List

Traversing a list is a technique to access an individual element of that list.

1. Using for loop for loop is used when you want to traverse each element of a list.

```
>>> a = ['p', 'r', 'o', 'g', 'r', 'a', 'm']
>>> for x in a:
    print(x, end = ' ')
output : p r o g r a m
```

2. Using for loop with range()

```
>>> a = ['p', 'r', 'o', 'g', 'r', 'a', 'm']
>>> for x in range(len(a)):
    print(x, end = ' ')
output : p r o g r a m
```

## List Operations

### 1. Concatenate Lists

List concatenation is the technique of combining two lists . The use of + operator can easily add the whole of one list to other list . Syntax list list1 + list2 e.g.

```
>>> L1 = [ 43, 56 , 34 ]
>>> L2 = [ 22 , 34 , 98 ]
>>> L = L1 + L2
>>> L [ 43, 56, 34, 22 , 34 , 98 ]
```

### 2. Replicating List

Elements of the list can be replicated using \* operator .

```
Syntax list = listl * digit e.g.
>>> L1 = [ 3 , 2 , 6 ]
>>> L = L1 * 2
>>> L [ 3 , 2 , 6 , 3 , 2 , 6 ]
```

3. **Slicing of a List:** List slicing refers to access a specific portion or a subset of the list for some operation while the original list remains unaffected .

```
Syntax:- list_name [ start: end ]
>>> List1 = [ 4 , 3 , 7 , 6 , 4 , 9 , 5, 0, 3 , 2 ]
>>> S = List1[ 2 : 5 ]
>>> S
[ 7, 6, 4 ]
Syntax: list_name [ start: stop : step ]
>>> List1 = [ 4 , 3 , 7 , 6 , 4 , 9 , 5, 0, 3 , 2 ]
>>> S = List1[ 1 : 9 : 3 ]
>>> S
[ 3, 4, 0 ]
```

## List Manipulation

### Updating Elements in a List

List can be modified after it created using slicing e.g.

```
>>> l1 = [ 2 , 4 , " Try " , 54 , " Again " ]
>>> l1[ 0 : 2 ] = [ 34 , " Hello " ]
>>> l1
[34, 'Hello', 'Try', 54, 'Again']
>>> l1[ 4 ] = [ "World " ]
>>> l1
[34, 'Hello', 'Try', 54, ['World']]
```

### Deleting Elements from a List

del keyword is used to delete the elements from the list .

Syntax:-

```
del list_name [ index ] # to delete individual element
del list_name [ start : stop ] # to delete elements in list slice c.g.
>>> list1 = [ 2.5 , 4 , 7 , 7 , 7 , 8 , 90 ]
>>> del list1[3]
>>> list1
[2.5, 4, 7, 7, 8, 90]
>>> del list1[2:4]
>>> list1
[2.5, 4, 8, 90]
```

## BUILT IN LIST METHODS

Method	Description
len()	Returns the length of the list passed as the argument
list()	Creates an empty list if no argument is passed Creates a list if a sequence is passed as an argument

append()	Appends a single element passed as an argument at the end of the list
extend()	Appends each element of the list passed as argument to the end of the given list
insert()	Inserts an element at a particular index in the list
count()	Returns the number of times a given element appears in the list
index()	Returns index of the first occurrence of the element in the list. If the element is not present, ValueError is generated
remove()	Removes the given element from the list. If the element is present multiple times, only the first occurrence is removed. If the element is not present, then ValueError is generated
pop()	Returns the element whose index is passed as parameter to this function and also removes it from the list. If no parameter is given, then it returns and removes the last element of the list
reverse()	Reverses the order of elements in the given list
sort()	Sorts the elements of the given list in-place
sorted()	It takes a list as parameter and creates a new list consisting of the same elements arranged in sorted order
min()	Returns minimum or smallest element of the list
max()	Returns maximum or largest element of the list
sum()	Returns sum of the elements of the list

## TUPLES

A tuple is an ordered sequence of elements of different data types. Tuple holds a sequence of heterogeneous elements, it store a fixed set of elements and do not allow changes

### Tuple vs List

Elements of a tuple are immutable whereas elements of a list are mutable.

Tuples are declared in parentheses ( ) while lists are declared in square brackets [ ].

Iterating over the elements of a tuple is faster compared to iterating over a list.

### **Creating a Tuple**

To create a tuple in Python, the elements are kept in parentheses ( ), separated by commas.

```
a = ( 34 , 76 , 12 , 90 )
```

```
b = ( 's ' , 3 , 6 , 'a ' )
```

- Accessing tuple elements, Traversing a tuple, Concatenation of tuples, Replication of tuples and slicing of tuples works same as that of List

### BUILT IN TUPLE METHODS

Method	Description
len()	Returns the length or the number of elements of the tuple passed as Argument
tuple()	Creates an empty tuple if no argument is passed. Creates a tuple if a sequence is passed as argument
count()	Returns the number of times the given element appears in the tuple
index()	Returns the index of the first occurrence of a given element in the tuple
sorted()	Takes elements in the tuple and returns a new sorted list. It should be noted that, sorted() does not make any change to the original tuple
min()	Returns minimum or smallest element of the tuple
max()	Returns maximum or largest element of the tuple
sum()	Returns sum of the elements of the tuple

## DICTIONARY

Dictionary is an unordered collection of data values that store the key : value pair instead of single value as an element . Keys of a dictionary must be unique and of immutable data types such as strings, tuples etc. Dictionaries are also called **mappings** or **hashes** or **associative arrays**

### Creating a Dictionary

To create a dictionary in Python, key value pair is used .

Dictionary is list in curly brackets , inside these curly brackets , keys and values are declared .

Syntax dictionary\_name = { key1 : value1 , key2 : value2 ... } Each key is separated from its value by a colon ( : ) while each element is separated by commas .

```
>>> Employees = { " Abhi " : " Manger " , " Manish " : " Project Manager " , " Aasha " : " Analyst " , " Deepak " : " Programmer " , " Ishika " : " Tester " }
```

### Accessing elements from a Dictionary

Syntax: dictionary\_name[keys]

```
>>> Employees[' Aasha ']  
' Analyst '
```

### Traversing a Dictionary

#### 1. Iterate through all keys

```
>>> for i in Employees:  
    print(i)
```

Output:

```
Abhi  
Manish  
Aasha  
Deepak  
Ishika
```

#### 3. Iterate through key and values

```
>>> for i in Employees:  
    print(i, " : ", Employees[i])
```

Output:

```
Abhi : Manger  
Manish : Project Manager  
Aasha : Analyst  
Deepak : Programmer  
Ishika : Tester
```

### Adding elements to a Dictionary

Syntax: dictionary\_name[new\_key] = value

```
>>> Employees['Neha'] = "HR"
```

```
>>> Employees
```

```
{' Abhi ' : ' Manger ' , ' Manish ' : ' Project Manager ' , ' Aasha ' : ' Analyst ' , ' Deepak ' : ' Programmer ' , ' Ishika ' : ' Tester ' , ' Neha ' : ' HR ' }
```

### Updating elements in a Dictionary

Syntax: dictionary\_name[existing\_key] = value

```
>>> Employees['Neha'] = " Progammer "
```

```
>>> Employees
```

```
{' Abhi ' : ' Manger ' , ' Manish ' : ' Project Manager ' , ' Aasha ' : ' Analyst ' , ' Deepak ' : ' Programmer ' , ' Ishika ' : ' Tester ' , ' Neha ' : ' Progammer ' }
```

### Membership operators in Dictionary

Two membership operators are **in** and **not in**. The membership operator inchecks if the key is present in the dictionary

```
>>> " Ishika " in Employees
```

```
True
```

#### 2. Iterate through all values

```
>>> for i in Employees:  
    print(Employees[i])
```

Output:

```
Manger  
Project Manager  
Analyst  
Programmer  
Tester
```

#### 4. Iterate through key and values simultaneously

```
>>> for a,b in Employees.items():  
    print("Key = ",a," and respective value = ",b)
```

Output:

```
Key = Abhi  and respective value = Manger  
Key = Manish  and respective value = Project Manager  
Key = Aasha  and respective value = Analyst  
Key = Deepak  and respective value = Programmer  
Key = Ishika  and respective value = Tester
```

```
>>> ' Analyst ' not in Employees
```

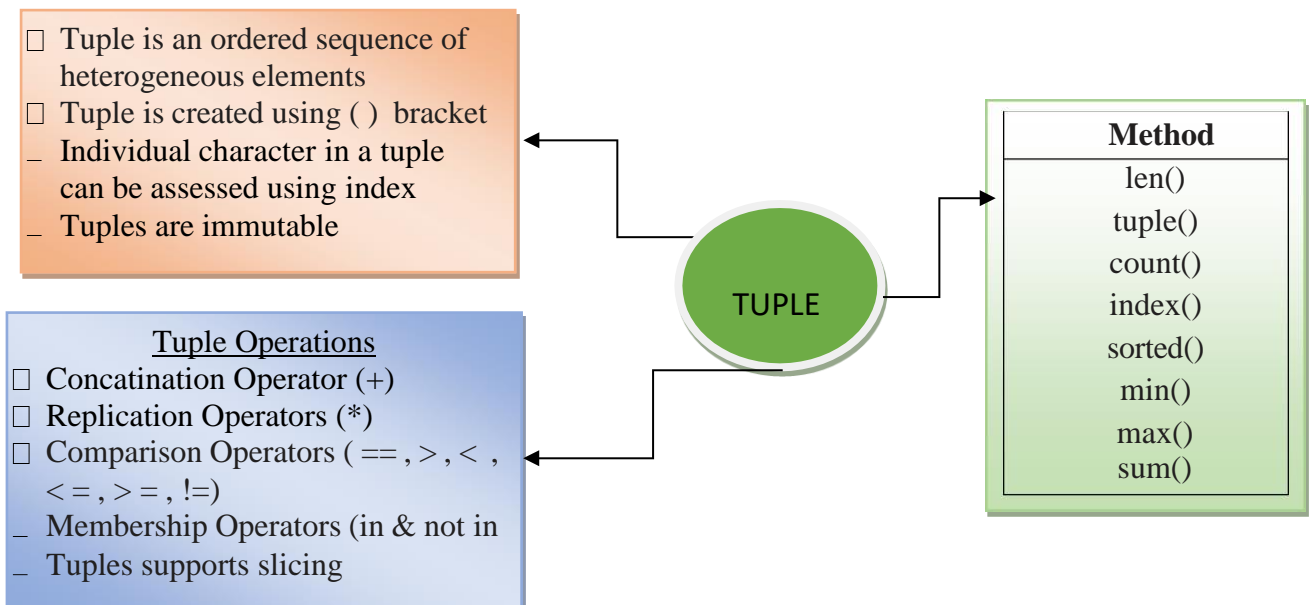
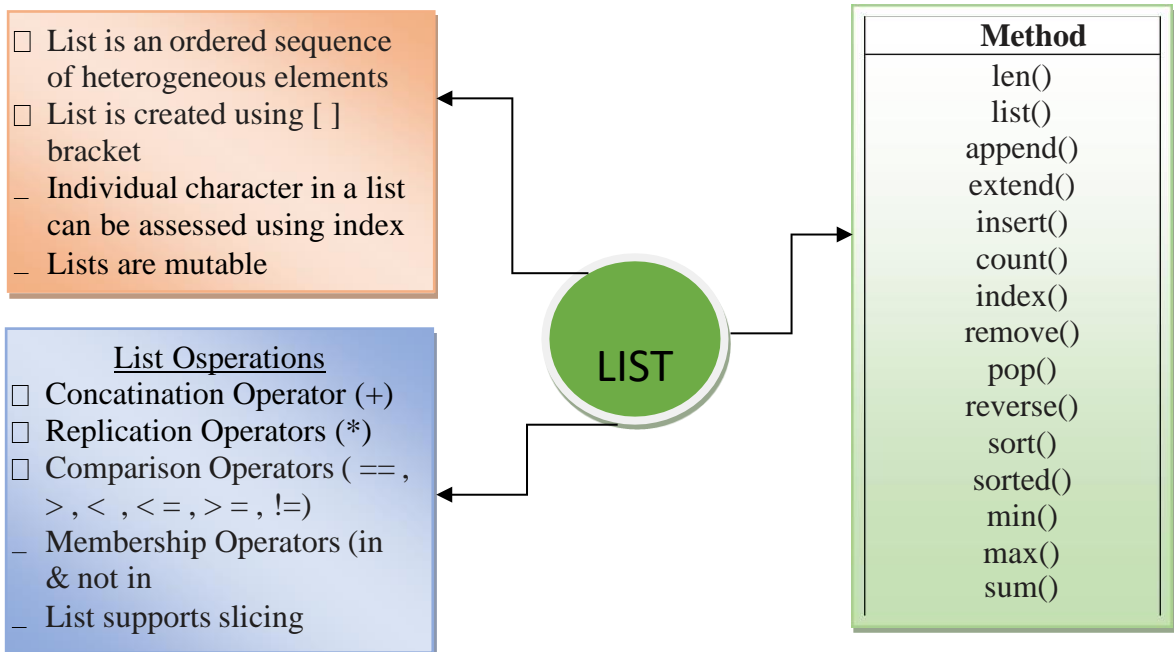
```
True
```



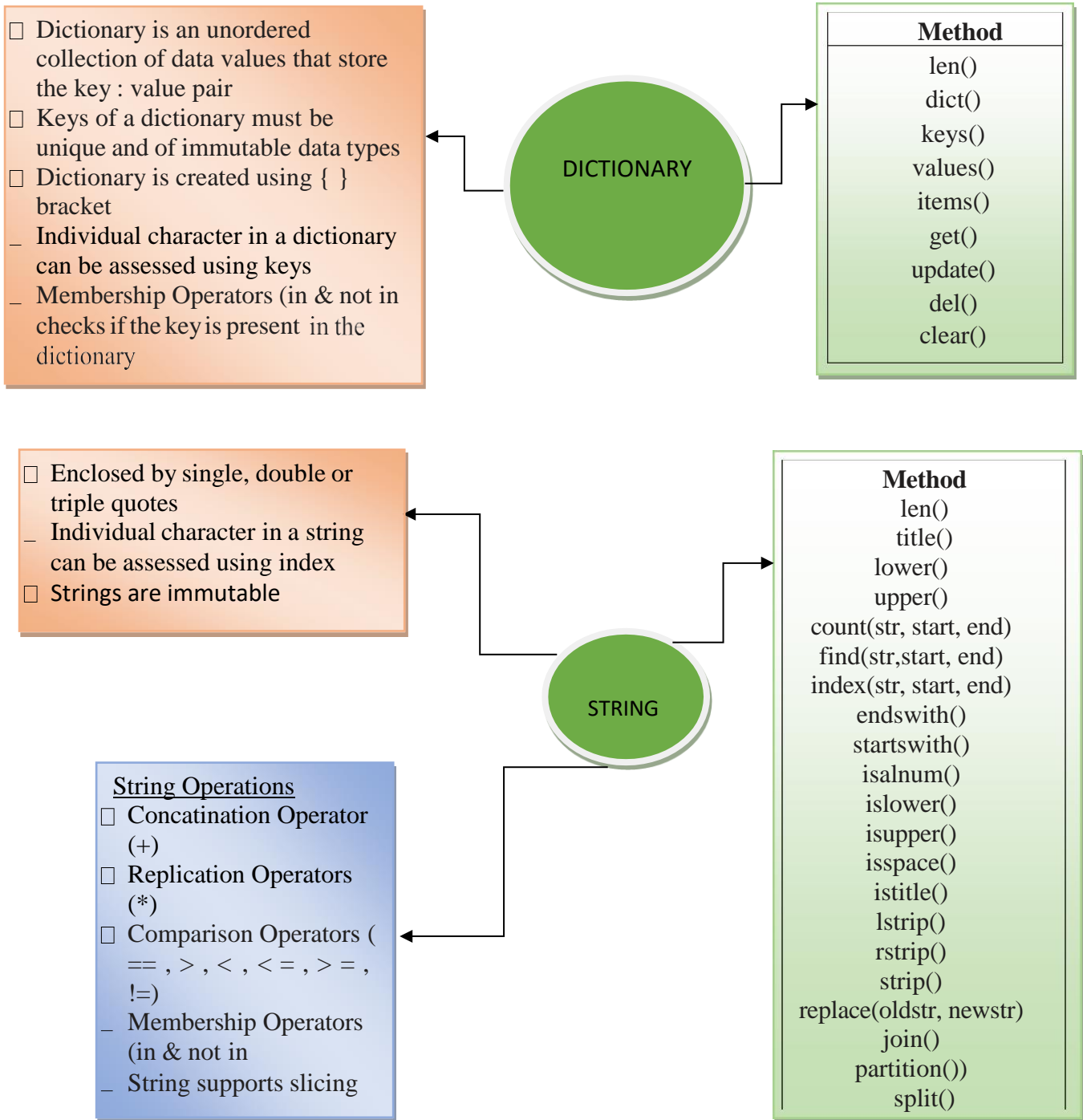
## BUILT IN DICTIONARY METHODS

Method	Description
len()	Returns the length or number of key: value pairs of the dictionary
dict()	Creates a dictionary from a sequence of key-value pairs
keys()	Returns a list of keys in the dictionary
values()	Returns a list of values in the dictionary
items()	Returns a list of tuples(key – value) pair
get()	Returns the value corresponding to the key passed as the argument If the key is not present in the dictionary it will return None
update()	appends the key-value pair of the dictionary passed as the argument to the key-value pair of the given dictionary
del()	Deletes the item with the given key To delete the dictionary from the memory we write: del Dict_name
clear()	Deletes or clear all the items of the dictionary

### MIND MAP



## MIND MAP



## QUESTIONS:

### 1 MARK QUESTIONS

1. What will be the output of the following set of commands

```
>>> str = "hello"
>>> str[:2]
```

- a. lo            b. he            c. llo            d. el

2. Which type of object is given below

```
>>> L = 1,23,"hello",1
```

- a. list            b. dictionary    c. array            d. tuple

3. Which operator tells whether an element is present in a sequence or not

- a. exist            b. in            c. into            d. inside

4. Suppose a tuple T is declared as T = (10,12,43,39), which of the following is incorrect  
 a. print(T[1])      b. T[2] = -2      c. print(max(T))      d. print(len(T))
5. Which index number is used to represent the last character of a string  
 a. -1      b. 1      c. n      d. n - 1
6. Which function returns the occurrence of a given element in a list?  
 a. len()      b. sum()      c. extend()      d. count()
7. which type of slicing is used to print the elements of a tuple in reverse order  
 a.[:-1]      b.[: :-1]      c. [1 : :]      d. [: : 1]
8. Dictionaries are also called  
 a. mapping      b. hashes      c. associative array      d. all of these
9. Which function returns the value of a given key, if present, from a dictionary?  
 a. items()      b. get()      c. clear()      d. keys()
10. The return type of input() function is:  
 a. list      b. integer      c. string      d. tuple

### ANSWERS

1	B	6	d
2	D	7	b
3	B	8	d
4	B	9	b
5	A	10	c

### 2 MARKS QUESTIONS

Q1. Rewrite the following code in python after removing all syntax error(s). Underline each correction done in the code.

```
STRING=""WELCOME
NOTE""
for S in range[0,8]:
print (STRING(S))
```

Q2. Find output generated by the following code:

```
Str="Computer"
Str=Str[-4:]
print(Str*2)
```

Q3. What will be the output of the following question

```
L = [10,19,45,77,10,22,2]
i) L.sort()      ii) max(L)
print(L)
```

Q4. Find the output

```
L = [10,19,45,77,10,22,2]
i) L[3:5]      ii) L[: :-2]
```

Q5. Distinguish between list and tuple.

Q6. Read the code given below and show the keys and values separately.

```
D = {'one' : 1, 'two' : 2, 'three' : 3}
```

Q7. Observe the given list and answer the question that follows.

```
List1 = [23,45,63, 'hello', 20, 'world',15,18]
i) list1[-3]      ii) list1[3]
```

Q8. Assertion (A) :

```
s = [11, 12, 13, 14]
s[1] = 15
```

Reasoning (R) : List is immutable.

- (A) Both A and R are true and R is the correct explanation of assertion.  
 (B) A and R both are true but R is not the correct explanation of A .  
 (C) A is true, R is false.  
 (D) A is false, R is true.

Q9. a=(1,2,3)  
a[0]=4

Assertion: The above code will result in error

Reason: Tuples are immutable. So we can't change them.

- (A) Both Assertion and reason are true and reason is correct explanation of assertion.
- (B) Assertion and reason both are true but reason is not the correct explanation of assertion.
- (C) Assertion is true, reason is false.
- (D) Assertion is false, reason is true.

### ANSWERS

Q1. CORRECTED CODE:-

STRING= "WELCOME"

NOTE=" "

for S in range (0, 7) :

print (STRING [S])

Also range(0,8) will give a runtime error as the index is out of range. It should be range(0,7)

Q2.

uter

'ComputerComputer'

Q5.

Q3.

[2, 10, 10, 19, 22, 45, 77]

Q4.

[77, 10]

[2, 10, 45, 10]

List	Tuple
Elements of a list are mutable	Elements of tuple are immutable
List is declared in square brackets []	Tuple is declared in parenthesis ()
Iterating over elements in list is slower as compared to tuple	Iterating over elements of tuples is faster as compared to list
e.g L1 = [1,2,3]	e.g T1 = (1,2,3)

Q6.

Keys: 'one', 'two', 'three'

Values: 1,2,3

Q7.

'world'

'hello'

Q8. (C)

Q9. (A)

### 3 MARKS QUESTIONS

Q1. Which of the string built in methods are used in following conditions?

- ii) Returns the length of a string
- iii) Removes all leading whitespaces in string
- iv) Returns the minimum alphabetic character from a string

Q2. Write a program to remove all the characters of odd index value in a string

Q3. Write a python program to count the frequencies of each elements of a list using dictionary

Q4. what will be the output of the following python code

L = [10,20]

L1 = [30,40]

L2 = [50,60]

L.append(L1)

print(L)

L.extend(L2)

print(L)

print(len(L))

Q5. Find the output of the given question

t = (4,0,'hello',90,'two',('one',45),34,2)

- i) t[5]
- ii) t[3:7]
- iii) t[1] + t[-2]

## ANSWERS

- Q1. i) len()                    ii) lstrip()                    iii) min()
- Q2.     str = input("Enter a string ")  
       final = ""  
       For i in range(len(str)):  
           if (i%2 == 0):  
               final = final + str[i]  
       print("The modified string is ",final)
- Q3.     L1 = []  
       n = int(input("Enter number of elements of the list "))  
       for i in range(0,n):  
           ele = int(input())  
       L1.append(ele)  
       print("Original list = ",L1)  
       print("Elements of list with their frequencies : ")  
       freq = {}  
       for item in L1:  
           if item in freq:  
               freq[item] += 1  
           else:  
               freq[item] = 1  
       for k,v in freq.items():  
           print("Element", k, "frequency", v)
- Q4.     [10, 20, [30, 40]]  
       [10, 20, [30, 40],50,60]
- Q5.     i) ('one',45)  
       ii) (90,'two',('one',45),34)  
       iii) 34

## 4 MARKS QUESTIONS

- Q1. Find the output
- i) 'python'.capitalize()
  - ii) max('12321')
  - iii) 'python'.index('ho')
  - iv) 'python'.endswith('thon')
- Q2. Consider the following code and answer the question that follows.
- ```
book = { 1:'Thriller',2:'Mystery',3:'Crime',4:'Children Stories' }  
library = { 5:'Madras Diaries',6:'Malgudi Days' }
```
- v) Ramesh wants to change the book 'Crime' to 'Crime Thriller'. He has written the following code:
- ```
book['Crime'] = 'Crime Thriller'
```
- but he is not getting the answer. Help him to write the correct command.
- vi) Ramesh wants to merge the dictionary book with the dictionary library. Help him to write the command.
- Q3. Write the suitable method names for the conditions given below:
- i) Add an element at the end of the list
  - ii) Return the index of first occurrence of an element
  - iii) Add the content of list2 at the end of list1
  - iv) Arrange the elements of a list1 in descending order

## ANSWERS

- Q1.  
i) 'Python'     ii) '3'                    iii) 3                    iv) True

- Q2. i) book[3] = 'Crime Thriller'  
 ii) library.update(book)
- Q3. i) append()                    ii) index()                    iii) list1.extend(list2)  
 iv) list1.sort(reverse = True)

### 5 MARKS QUESTIONS

Q1. Find the output of the following code:

```
a = (5,(7,5,(1,2)),5,4)
print(a.count(5))
print(a[1][2])
print(a * 3)
print(len(a))
b = (7,8,(4,5))
print(a + b)
```

Q2. Following is a program to check a list is same if it is read from front or from back. Observe the program and answer the following questions:

```
a = [1,2,3,3,2,1]
i = _____ # statement 1
mid = (len(a)) / 2
same = True
while _____: # statement 2
    if a[i] != _____: # statement 3
        print("NO")
        same = False
        break
    _____ # statement 4
if same == _____: # statement 5
    print("YES")
```

- vii) Which value will be assigned to the variable I in statement 1?  
 viii) Fill the blank line in statement 2.  
 ix) Fill the blank line in statement 3.  
 x) Fill the blank line in statement 4.  
 xi) Fill the blank line in statement 5.

Q3. Explain the following string functions with examples.

- i) title()                    ii) count()                    iii) find()                    iv) index()                    v) join()

### ANSWERS

Q1. 2  
 (1, 2)  
 (5, (7, 5, (1, 2)), 5, 4, 5, (7, 5, (1, 2)), 5, 4, 5, (7, 5, (1, 2)), 5, 4)  
 4  
 (5, (7, 5, (1, 2)), 5, 4, 7, 8, (4, 5))

Q2. i) 0  
 ii) i < mid  
 iii) a[i] != a[len(a) - i - 1]  
 iv) i = i + 1  
 v) True

Q3. i) title()  
 Returns the string with first letter of every word in the string in uppercase and rest in lowercase.

```
>>> str1 = 'hello WORLD!'
>>> str1.title()
'Hello World!'
```

#### ii) count( )

Returns number of times substring str occurs in the given string. If we do not give start index and end index then searching starts from index 0 and ends at length of the string.

```
>>> str1 = 'Hello World! Hello Hello'
>>> str1.count('Hello',12,25)
2
>>> str1.count('Hello')
3
```

#### iii) find()

Returns the first occurrence of index of substring stroccurring in the given string. If we do not give start and end then searching starts from index 0 and ends at length of the string. If the substring is not present in the given string, then the function returns -1
---

```
>>> str1= 'Hello World! Hello Hello'
>>> str1.find('Hello',10,20)
13
>>> str1.find('Hello',15,25)
19
>>> str1.find('Hello')
0
>>> str1.find('Hee')
-1
```

#### iv) index( )

Same as find() but raises an exception if the substring is not present in the given string

```
>>> str1 = 'Hello World! Hello Hello'
>>> str1.index('Hello')
0
>>> str1.index('Hee')
ValueError: substring not found
```

#### v) join()

Returns a string in which the characters in the string have been joined by a separator

```
>>> str1 = ('HelloWorld!')
>>> str2 = '-' #separator
>>> str2.join(str1)
'H-e-l-l-o-W-o-r-l-d-!'
```

## FUNCTION IN PYTHON

**Functions:** types of function (built-in functions, functions defined in module, user defined functions), creating user defined function, arguments and parameters, default parameters, positional parameters, function returning value(s), flow of execution, scope of a variable (global scope, local scope)

Let us revise

- A function is a block of code that performs a specific task.
- Advantages of function: Reusability of code, Reduce size of code, minimum number of statements, minimum storage, Easy to manage and maintain
- Types of functions: Built-in-functions, Functions defined in module, User defined function
- Built-in functions are the functions whose functionality is pre-defined in python like abs(), eval(), input(), print(), pow()
- Some functions are defined inside the module like load() and dump() function defined inside the pickle module.
- A function that can be defined by the user is known as user defined function.
- def keyword is used to define a function.
- There is a colon at the end of def line, meaning it requires block
- User Defined function involved two steps:  
defining  
calling
- **Syntax for user defined function:**  
def <function name>( [parameter list ]):  
    ["""function's doc string """]  
    <statement>  
    [<statement>]
- Python supports three types of formal arguments/ parameters: Positional Arguments, Default parameters, Keyword (or named ) Arguments
- **Positional Arguments:** When the function call statement must match the number and order of arguments as defined in the function definition, this is called the positional argument matching.
- A parameter having default value in the function header is known as a **default parameter**.
- **Keyword Arguments** are the named arguments with assigned values being passed in the function call statement.
- A function may or may not return one or more values.
- A function that does not return a value is known as void function and returns legal empty value None.
- Functions returning value are also known as fruitful functions.
- The flow of execution refers to the order in which statements are executed during a program.
- A variable declared in a function body (block) is said to have local scope. i.e. it can be accessed within this function.
- A variable declared outside of all functions/top level of segment of a program is said to have global scope. i.e. it can be accessible in whole program and all blocks ( functions and the other blocks contained within program.



# MIND MAP ON FUNCTION

**Function:** A function is a group of statements that exists within a program for the purpose of performing a specific task.

## Types

**Scope:** Scope of a variable is the area of the program where it may be referenced.

### **Built-in-functions:**

Built-in functions are the predefined functions that are already available in the python. Ex- int(),

### **Function defined in module:**

A module is a file containing functions and variables defined in separate files.

### **User Defined functions:**

A function is a block of code which only runs when it is called. In Python, a function is defined using the def keyword.

**Local:** A variable created inside a function belongs to the local scope of that function, and can only be used inside that function.

Ex-

```
x = 300
def myfunc():
    print(x)
myfunc()
print(x)
```

**Global:** A variable declared outside of all functions/top level of segment of a program is said to have global scope. i.e. it can be accessible in whole program and all blocks.

Ex-

```
def myfunc():
    x = 300
    print(x)
myfunc()
```

## Types of Arguments/Parameters

### **Positional Arguments:**

Arguments passed to a function in correct positional order.

### **Default Arguments:**

A default argument is an argument that assumes a default value if a value is not provided in the function call for that argument.

### **Keyword Arguments:**

Value can be provided by using their name instead of the position (order) in function call statement. These are called keyword arguments.

### Multiple Choice Questions (1 Mark)

1. What is the default return value for a function that does not return any value explicitly?  
(a) None (b) int (c) double (d) null
2. Which of the following items are present in the function header?  
(a) Function name only (b) Parameter list only  
(c) Both function name and parameter list (d) return value
3. Which of the following keyword marks the beginning of the function block?  
(a) func (b) define (c) def (d) function
4. Pick one of the following statements to correctly complete the function body in the given code snippet.  
def f(number):  
    # Missing function body  
    print (f(5))  
(a) return "number" (b) print(number) (c) print("number") (d) return number
5. Which of the following function header is correct?  
(a) def f(a=1,b): (b) def f(a=1,b,c=2):  
(c) def f(a=1, b=1, c=2): (d) def f(a=1,b=1,c=2,d);
6. Which of the following statements is not true for parameter passing to functions?  
(a) You can pass positional arguments in any order.  
(b) You can pass keyword arguments in any order.  
(c) You can call a function with positional and keyword arguments.  
(d) Positional arguments must be before keyword arguments in a function call
7. A variable defined outside all the functions referred to as.....  
(a) A static variable (b) A global variable (c) A local variable (d) An automatic variable
8. What is the order of resolving scope of a name in a python program?  
(L: Local namespace, E: Enclosing namespace, B: Built-In namespace, G: Global namespace)  
(a) BGEL (b) LEGB (c) GEBL (d) LBEG
9. Assertion (A):- If the arguments in a function call statement match the number and order of arguments as defined in the function definition, such arguments are called positional arguments.  
Reasoning (R):- During a function call, the argument list first contains default argument(s) followed by positional argument(s).  
(a) Both A and R are true and R is the correct explanation for A  
(b) Both A and R are true and R is not the correct explanation for A  
(c) A is True but R is False  
(d) A is false but R is True
10. The .....refers to the order in which statements are executed during a program run.  
(a) Token (b) Flow of execution (c) Iteration (d) All of the above
11. Choose the correct option:  
Statement1: Local Variables are accessible only within a function or block in which it is declared.  
Statement2: Global variables are accessible in the whole program.  
(a) Statement1 is correct but Statement2 is incorrect  
(b) Statement2 is correct but Statement1 is incorrect  
(c) Both Statements are Correct  
(d) Both Statements are incorrect
12. The .....of a variable is the area of the program where it may be referenced  
(a) external (b) global (c) scope (d) local

## Answers

Questions	Answers
1.	(a) None
2.	(c) Both function name and parameter list
3.	(c) def
4.	(d) return number
5.	(c) def f(a=1, b=1, c=2):
6.	(a) You can pass positional arguments in any order.
7.	(b) A global variable
8.	(b) LEGB
9.	(c) A is True but R is False
10.	(b) Flow of execution
11.	(c) Both Statements are Correct
12.	c) scope

### Short Questions (2 Marks)

Q1. What do you mean by a function? How is it useful?

**Answer:** A function is a block of code that performs a specific task. Functions are useful as they can be reused anywhere through their function call statements.

Q2. Observe the following Python code very carefully and rewrite it after removing all syntactical errors with each correction underlined.

```
def execmain():
    x = input("Enter a number:")
    if (abs(x)= x):
        print("You entered a positive number")
    else:
        x=-1
    print("Number made positive : ",x)
execmain()
```

**Answer:**

```
def execmain():
    x = int(input("Enter a number:"))
    if (abs(x)== x):
        print("You entered a positive number")
    else:
        x*=-1
    print("Number made positive : ",x)
execmain()
```

Q3. What is an argument? Give an example.

**Answer:** An argument is data passed to a function through function call statement. It is also called actual argument or actual parameter. For example, in the statement `print(math.sqrt(25))`, the integer 25 is an argument.

Q4. What is the output of the program given below?

```
x = 75
def func (x) :
    x = 10
func (x)
print ('x is now', x)
```

**Answer:** x is now 75

Q5. What will be the output of the following code:

```

total=0
def add(a,b):
    global total
    total=a+b
    print(total)
add(6,6)
print(total)

```

**Answer:** 12  
12

Q6. Is return statement optional? Compare and comment on the following two return statements:

- (i) return
- (ii) return val

**Answer:** The return statement is optional only when the function does not return a value. A function that returns a value must have at least one return statement.

From the given two return statements,

- (i) The statement return is not returning any value. Rather it returns the control to caller along with empty value None.
- (ii) The statement return val is returning the control to caller along with the value contained in variable val.

Q7. Divyansh, a python programmer, is working on a project which requires him to define a function with name CalculateInterest(). He defines it as:

```
def CalculateInterest(Principal,Rate=.06, Time): # Code
```

But this code is not working, Can you help Divyansh to identify the error in the above function and with the solution?

**Answer.** Yes, here non-default argument is followed by default argument which is wrong as per python's syntax. (while passing default arguments to a function ,all the arguments to its right must also have default values, otherwise it will result in an error.)

Q8. Write a function that takes a positive integer and returns the one's position digit of the integer.

**Answer:**

```
def getOnes(num):
    oneDigit=num%10 # return the ones digit of the integer num
    return oneDigit
```

Q9. Anita has written a code to input a number and check whether it is prime or not. His code is having errors. Rewrite the correct code and underline the corrections made.

```

def prime():
    n=int(input("Enter number to check :: "))
    for i in range (2, n//2):
        if n%i=0:
            print("Number is not prime \n")
            break
        else:
            print("Number is prime \n")

```

**Answer:**

```

def prime():
    n=int(input("Enter number to check :: ")) #bracket missing
    for i in range (2, n//2):
        if n%i==0: # = missing
            print("Number is not prime \n")
            break #wrong indent
        else:
            print("Number is prime \n") # quote mismatch

```

Q10. What is the difference between parameter and argument?

**Answer:**

Parameters are temporary variable names within functions. The argument can be thought of as the

value that is assigned to that temporary variable.

For instance, let's consider the following simple function to calculate sum of two numbers.

```
def sum(a,b):  
    return a+b  
sum(10,20)
```

Here a, b are the parameters for the function 'sum'. Arguments are used in procedure calls, i.e., the values passed to the function at runtime. 10, 20 are the arguments for the function sum.

Q11. What is the significance of having functions in a program?

**Answer:** Creating functions in programs is very useful. It offers the following advantages:

- (i) The program is easier to understand.
- (ii) Redundant code is at one place, so making changes is easier.
- (iii) Reusable functions can be put in a library in modules.

Q12. What is the difference between local variable and global variable? Also give a suitable Python code

to illustrate both.

Answer:

S.No.	Local Variable	Global Variable
1.	It is a variable which is declared within a function or within a block.	It is a variable which is declared outside all the functions or in a global space.
2.	It cannot be accessed outside the function but only within a function/block of a program.	It is accessible throughout the program in which it is declared.

For example, in the following code x, xcubed are global variables and n and cn are local variables.

```
def cube(n):    # n and cn are local variables  
    cn=n*n*n  
    return cn  
x=10           # x is a global variable  
xcubed=cube(x) # xcubed is a global variable  
print(x, "Cubed 15", xcubed)
```

### Short Questions (3 Marks)

Q1. Consider the following function that takes two positive integer parameters a and b. Answer the following questions based on the code below:

```
def funct1(a,b):  
    if a>1:  
        if a%b==0:  
            print (b, end=' ')  
            funct1(int(a/b),b)  
        else:  
            funct1(a,b+1)
```

- (a) What will be printed by the function call funct1(24,2)?
- (b) What will be printed by the function call funct1(84,2)?

(c) State in one line what `funct1()` is trying to calculate.

**Answer** (a) 2 2 2 3

(b) 2 2 3 7

(c) finding factors of A which are greater than and equal to B.

Q2. Write a user defined function to print the odd numbers from a given list passed as an argument.

Sample list: [1,2,3,4,5,6,,7,8,9,10]

Expected Result: [1,3,5,7,9]

**Answer:**

```
def odd_num(l):
    odd=[]
    for n in l:
        if n%2!=0:
            odd.append(n)
    return odd
print(odd_num([1,2,3,4,5,6,7,8,9,10]))
```

Q3. Which line in the given code(s) will not work and why?

```
def interest(p,r,t=7):
    I=(p*r*t)
    print(interest(20000,.08,15))           #line 1
    print(interest(t=10, 20000, 0.75))     #line 2
    print(interest(50000, 0.7))           #line 3
    print(interest(p=10000,r=.06,time=8))  #line 4
    print(interest(80000, t=10))          #line 5
```

**Answer:** Line 2: positional argument must not be followed by keyword argument, i.e., positional argument

must appear before a keyword argument.

Line 4: There is no keyword argument with name 'time'

Line 5: Missing value for positional arguments, R.

Q4. Write a python function `showlarge()` that accepts a string as parameter and prints the words whose length is more than 4 characters.

Eg: if the given string is "My life is for serving my Country"

The output should be

serving

Country

**Answer:**

```
def showlarge(s):
    l = s.split()
    for x in l:
        if len(x)>4:
            print(x)
s=" My life is for serving my Country "
showlarge(s)
```

Q5. Write a function `Interchange (num)` in Python, which accepts a list num of integers, and interchanges the adjacent elements of the list and print the modified list as shown below:

(Number of elements in the list is assumed as even)

Original List: num = [5,7,9,11,13,15]

After Rearrangement num = [7,5,11,9,15,13]

**Answer:**

```
def Interchange(num):
    for i in range(0,n,2):
```

```

    num[i], num[i+1] = num[i+1], num[i]
    print(num)
num=[5,7,9,11,13,15]
n=len(num)
if n%2==0:
    Interchange(num)

```

Q6. Write a function INDEX\_LIST(L), where L is the list of elements passed as argument to the function. The function returns another list named 'indexList' that stores the indices of all Non-Zero Elements of L.

For example:

If L contains [12, 4, 0, 11,0, 56]

The indexList will have - [0,1,3,5]

**Answer:**

```

def INDEX_LIST(L):
    indexList=[]
    for i in range(len(L)):
        if L[i]!=0:
            indexList.append(i)
    return indexList

```

### Long Questions/Case Study/Application Based (4 Marks)

**Q1. Krishnav is looking for his dream job but has some restrictions. He loves Delhi and would take a job there if he is paid over Rs.40,000 a month. He hates Chennai and demands at least Rs. 1,00,000 to work there. In any another location he is willing to work for Rs. 60,000 a month. The following code shows his basic strategy for evaluating a job offer.**

**def DreamProject():**

**pay= \_\_\_\_\_**

**location= \_\_\_\_\_**

**if location == "Mumbai":**

**print ("I'll take it!") #Statement 1**

**elif location == "Chennai":**

**if pay < 100000:**

**print ("No way") #Statement 2**

**else:**

**print("I am willing!") #Statement 3**

**elif location == "Delhi" and pay > 40000:**

**print("I am happy to join") #Statement 4**

**elif pay > 60000:**

**print("I accept the offer") #Statement 5**

**else:**

**print("No thanks, I can find something better") #Statement 6**

On the basis of the above code, choose the right statement which will be executed when different inputs for pay and location are given

(i) Input: location = "Chennai", pay = 50000

a. Statement 1                      b. Statement 2                      c. Statement 3                      d. Statement 4

(ii) Input: location = "Surat", pay = 50000

a. Statement 2                      b. Statement 4                      c. Statement 5                      d. Statement 6

(iii) Input- location = "Any Other City", pay = 1

a Statement 1                      b. Statement 2                      c. Statement 4                      d. Statement 6

(iv) Input location = "Delhi", pay = 500000

a. Statement 6

b. Statement 5

c. Statement 4

d. Statement 3

**Answer:** (i) b. Statement2 (ii) d. Statement6 (iii) d. Statement6 (iv) c. Statement4

Q2. Kids Elementary is a Playway school that focuses on "Play and learn" strategy that helps toddlers understand concepts in a fun way. Being a senior programmer, you have taken responsibility to develop a program using user-defined functions to help children differentiate between upper case and lower case letters/ English alphabet in a given sequence. Make sure that you perform a careful analysis of the type of alphabets and sentences that can be included as per age and curriculum. Write a python program that accepts a string and calculates the number of upper case letters and lower case letters.

**Answer:**

```
def string_test(s):
    d={"UPPER_CASE":0,"LOWER_CASE":0}
    for c in s:
        if c.isupper():
            d["UPPER_CASE"]+=1
        elif c.islower():
            d["LOWER_CASE"]+=1
        else:
            pass
    print("Original String:",s)
    print("No. of Upper Case Characters:",d["UPPER_CASE"])
    print("No. of Lower Case Characters:",d["LOWER_CASE"])
string_test("Play Learn and Grow")
```

### Case Based/ Source Based Integrated Questions (5 Marks)

Q1. Traffic accidents occur due to various reasons. While problems with roads or inadequate safety facilities lead to some accidents, majority of the accidents are caused by driver's carelessness and their failure to abide by traffic rules.

ITS Roadwork is a company that deals with manufacturing and installation of traffic lights so as to minimize the risk of accidents. Keeping in view the requirements, traffic simulation is to be done. Write a program in python that simulates a traffic light. The program should perform the following:

- (a) A user defined function trafficLight() that accepts input from the user, displays an error message if the following is displayed depending upon return value from light(); [2]
- (i) "STOP, Life is more important than speed", if the value returned by light() is 0.
  - (ii) "PLEASE GO SLOW", if the value returned by light() is 1.
  - (iii) "You may go now", if the value returned by light() is 2.
- (b) A user defined function light() that accepts a string as input and returns 0 when the input is RED, 1 when the input is YELLOW and 2 when the input is GREEN. The input should be passed as an argument. [2]
- (c) Display "BETTER LATE THAN NEVER" after the function trafficLight() is executed. [1]

**Answer:** (a)

#Program to simulate a traffic light comprising of two user defined functions trafficLight() and light()

```
def trafficLight():
    signal=input("Enter the colour of the traffic light:")
```



```

if signal not in ["RED", "YELLOW", "GREEN"]:
    print("Please enter a valid traffic light colour in CAPITALS")
else:
    value=light(signal)
    if value==0:
        print("STOP", "Life is more important than speed")
    elif value==1:
        print("PLEASE GO SLOW")
    else:
        print("You may go now")
(a) # function light()
def light(colour):
    if colour=="RED":
        return 0
    elif colour=="YELLOW":
        return 1
    else:
        return 2
trafficLight()

# Display BETTER LATE THAN NEVER
print("BETTER LATE THAN NEVER")

```

Q2. Observe the following code and select appropriate answers for the given questions:

```

total=1
def multiply(l):          # line1
    for x in l:
        _____total # line2
        total*=x
    return _____    # line3
l=[2,3,4]
print(multiply(_____), end=" ") # line4
print(" , Thank You")

```

- (i) Identify the part of function in # line1
  - (a) Function header
  - (b) Function calling
  - (c) Return statement
  - (d) Default argument
- (ii) Which of the keyword is used to fill in the blank for # line2 to run the program without error.
  - (a) eval
  - (b) def
  - (c) global
  - (d) return
- (iii) Which variable is going to be returned in # line3
  - (a) total
  - (b) x
  - (c) l
  - (d) None
- (iv) Which variable is required in the # line4
  - (a) total
  - (b) x
  - (c) l
  - (d) None
- (v) In the # line4 the multiple(l) is called
  - (a) Caller
  - (b) Called
  - (c) Parameters
  - (d) Arguments

**Answer:** (i) (a) Function header  
 (ii) (c) global  
 (iii) (a) total  
 (iv) (c) l  
 (v) (a) Caller

## Exception Handling

**Exception:** Contradictory or Unexpected situation or unexpected error, during program execution, is known as Exception.

**Exception Handling:** Way of handling anomalous situations in a program-run, is known as Exception Handling.

Some common examples of Exceptions are:

- Divide by zero errors
- Accessing the elements of an array beyond its range
- Invalid input
- Hard disk crash
- Opening a non-existent file
- Heap memory exhausted

Example:

try :

```
print ("result of 10/5 = ", (10/5))
print ("result of 10/0 = ", (10/0))
```

except :

```
print ("Divide by Zero Error! Denominator must not be zero!")
```

**The output produced by above code is as shown below :**

```
result of 10 / 5 = 2
result of 10 / 0 = Divide by Zero Error! Denominator must not be zero!
```

### 1 Marks Question:

1. Errors resulting out of violation of programming language's grammar rules are known as:  
(a) Compile time error (b) Logical error (c) Runtime error (d) Exception
2. An unexpected event that occurs during runtime and causes program disruption, is called:  
(a) Compile time error (b) Logical error (c) Runtime error (d) Exception
3. Which of the following keywords are not specific to exception handling ?  
(a) try (b) except (c) finally (d) else
4. Which of the following blocks is a 'must-execute' block ?  
(a) try (b) except (c) finally (d) else
5. Which keyword is used to force an exception ?  
(a) try (b) except (c) raise (d) finally

**Answers:**

1. (a) 2. (d) 3. (d) 4. (c) 5. (c)

**Predict the output of the following code for these function calls:**

(a) divide(2, 1)      (b) divide(2, 0)      (c) divide("2", "1")

```
def divide(x, y):
```

```
    try:
```

```
        result = x/y
```

```
    except ZeroDivisionError:
```

```
        print ("division by zero!")
```

```
    else:
```

```
        print ("result is", result)
```

```
    finally:
```

```
        print ("executing finally clause")
```

**Solution.**

(a) divide(2, 1) result is 2  
executing finally clause

(b) divide(2, 0) division by zero!  
executing finally clause

(c) divide("2", "1")  
executing finally clause

Traceback (most recent call last):

TypeError: unsupported operand type(s) for /: 'str' and 'str'

# FILE HANDLING – TEXT FILES

## I. INTRODUCTION:

- Files are named locations on disk to store related information. They are used to permanently store data in a non-volatile memory (e.g. hard disk).
- Since Random Access Memory (RAM) is volatile (which loses its data when the computer is turned off), we use files for future use of the data by permanently storing them.
- When we want to read from or write to a file, we need to open it first. When we are done, it needs to be closed so that the resources that are tied with the file are freed.
- Hence, in Python, a file operation takes place in the following order:
  - Open a file
  - Read or write (perform operation)
  - Close the file

## II Types of File in Python

There are two types of files in Python and each of them are explained below in detail with examples for your easy understanding. They are:

- 1) Binary file
- 2) Text file

## II. Text files in Python

- A text file is usually considered as sequence of lines. Line is a sequence of characters (ASCII), stored on permanent storage media. Although default character coding in python is ASCII but supports Unicode as well.
- In text file, each line is terminated by a special character, known as End of Line (EOL). From strings we know that `\n` is newline character.
- At the lowest level, text file is collection of bytes. Text files are stored in human readable form. They can also be created using any text editor.
- Text files don't have any specific encoding and it can be opened in normal text editor itself.

### Example of Text Files:

- **Web standards:** html, XML, CSS, JSON
- **Tabular data:** csv, tsv etc.
- **Configuration:** ini, cfg, reg etc
- **Source code:** c, app, js, py, java etc.
- **Documents:** txt, tex, RTF etc.

## III. OPENING OR CREATING A NEW FILE IN PYTHON

- The method `open()` is used to open an existing file or creating a new file. If the complete directory is not given then the file will be created in the directory in which the python file is stored. The syntax for using `open()` method is given below.
  - **Syntax:**  
– `file_object = open( file_name, "Access Mode", Buffering )`
- The open method returns file object which can be stored in the name **file object (file-handle)**.

**File name** is a unique name in a directory. The `open()` function will create the file with the specified name if it is not already exists otherwise it will open the already existing file.

### **File Access Modes:**

- **The access mode:** it is the string which tells in what mode the file should be opened for operations. There are three different access modes are available in python.
- **Reading: Reading mode** is crated only for reading the file. The pointer will be at the beginning of the file.
- **Writing: Writing mode** is used for overwriting the information on existing file.

- **Append:** *Append mode is same as the writing mode. Instead of over writing the information this mode append the information at the end.*
- Below is the list of representation of various access modes in python.

#### Access modes in Text Files

- 'r' – Read Mode: Read mode is used only to read data from the file.
- 'w' – Write Mode: This mode is used when you want to write data into the file or modify it. Remember write mode overwrites the data present in the file.
- 'a' – Append Mode: Append mode is used to append data to the file. Remember data will be appended at the end of the file pointer.
- 'r+' – Read or Write Mode: This mode is used when we want to write or read the data from the same file.
- 'a+' – Append or Read Mode: This mode is used when we want to read data from the file or append the data into the same file.

#### Buffering:

- Buffering is the process of storing a chunk of a file in a temporary memory until the file loads completely. In python there are different values can be given. If the buffering is set to 0 , then the buffering is off. The buffering will be set to 1 when we need to buffer the file.

#### Examples of Opening TEXT Files in Python

# open file in current directory

- `f = open("test.txt", "r")`

# specifying full path

- `f = open(r"D:\temp\data.txt", "r")` #-raw string
- `f = open("D:\\temp\\data.txt", "r")` #-absolute path

#### IV. CLOSING FILES IN PYTHON

- After processing the content in a file, the file must be saved and closed. To do this we can use another method `close()` for closing the file. This is an important method to be remembered while handling files in python.

- Syntax: `file_object.close()`  
`string = "This is a String in Python"`  
`my_file = open(my_file_name.txt,"w+",1)`  
`my_file.write(string)`  
`my_file.close()`  
`print(my_file.closed)`

#### V. READING INFORMATION IN THE FILE

- In order to read a file in python, we must open the file in read mode.
- There are three ways in which we can read the files in python.
  - `read([n])`
  - `readline([n])`
  - `readlines()` – all lines returned to a list

Here, n is the number of bytes to be read.

##### Example 1:

```
my_file = open("C:/Documents/Python/test.txt", "r")
print(my_file.read(5))
```

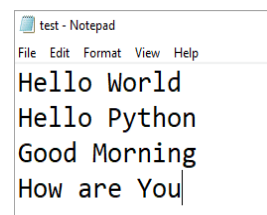
Output:

Hello

Here we are opening the file test.txt in a read-only mode and are reading only the first 5 characters of the file using the `my_file.read(5)` method.

##### Example 2:

```
my_file = open("C:/Documents/Python/test.txt", "r")
print(my_file.read())
```



Output:

Hello

Hello

Good Morning #Here we have not provided any argument inside the read() function.

#Hence it will read all the content present inside

the file.

**Example 3:**

```
my_file = open("C:/Documents/Python/test.txt", "r")
print(my_file.readline(2))
```

Output:

He

This function returns the first 2 characters of the next line.

**Example 4:**

```
my_file = open("C:/Documents/Python/test.txt", "r")
print(my_file.readline())
```

Output:

Hello World

Using this function we can read the content of the file on a line by line basis.

**Example 5:**

```
my_file = open("C:/Documents/Python/test.txt", "r")
print(my_file.readlines())
```

Output:

['Hello World\n', 'Hello Python\n', 'Good Morning']

Here we are reading all the lines present inside the text file including the newline characters.

**Example 5: Reading a specific line from a File**

```
line_number = 4
fo = open("C:/Documents/Python/test.txt", 'r')
currentline = 1
for line in fo:
    if(currentline == line_number):
        print(line)
        break
    currentline = currentline + 1
```

Output:

How are You

#In this example, we are trying to read only the 4<sup>th</sup> line from the 'test.txt' file using a "for loop".

**VI. WRITE TO A PYTHON TEXT FILE**

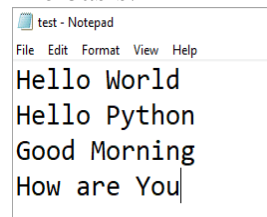
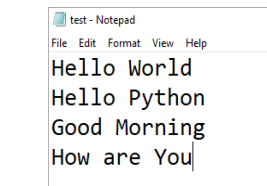
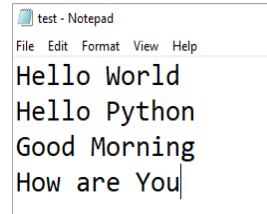
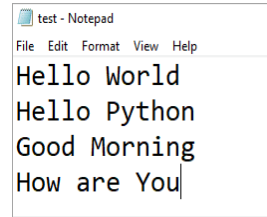
- In order to write data into a file, we must open the file in write mode.
- We need to be very careful while writing data into the file as it overwrites the content present inside the file that we are writing, and all the previous data will be erased.
- **We have two methods for writing data into a file as shown below.**

write(string) and writelines(list)

**Example 1:**

```
my_file = open("C:/Documents/Python/test.txt", "w")
my_file.write("Hello World")
```

World  
Python



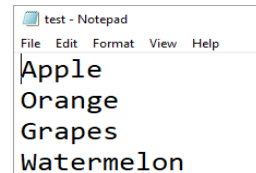
The above code writes the String 'Hello World' into the 'test.txt' file.

```
my_file = open("C:/Documents/Python/test.txt",  
w)  
my_file.write("Hello World\n")  
my_file.write("Hello Python")
```

- The first line will be 'Hello World' and as we have mentioned \n character, cursor will move to the next line of the file and then write 'Hello Python'.
- Remember if we don't mention \n character, then the data will be written continuously in the text file like 'Hello WorldHelloPython'

### **Example 3:**

```
fruits = ["Apple\n", "Orange\n", "Grapes\n", "Watermelon"]  
my_file = open("C:/Documents/Python/test.txt", "w")  
my_file.writelines(fruits)
```



The above code writes a **list of data** into the 'test.txt' file simultaneously.

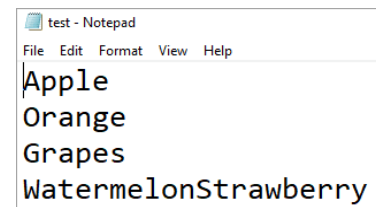
### **Append in a Python Text File:**

To append data into a file we must open the file in 'a+' mode so that we will have access to both the append as well as write modes.

### **Example 1:**

```
my_file = open("C:/Documents/Python/test.txt", "a+")  
my_file.write ("Strawberry")
```

The above code appends the string 'Strawberry' at the **end** of the 'test.txt' file

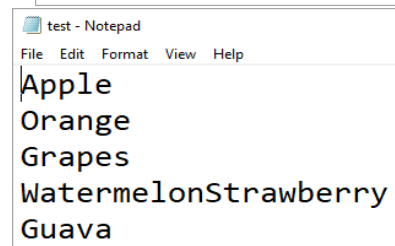


### **Example 2:**

```
my_file = open("C:/Documents/Python/test.txt", "a+")  
my_file.write ("\nGuava")
```

The above code appends the string 'Apple' at the end of

the 'test.txt' file **in a new line**



### **flush() function**

- When we write any data to file, python hold everything in buffer (temporary memory) and pushes it onto actual file later. If you want to force Python to write the content of buffer onto storage, you can use flush() function.
- Python automatically flushes the files when closing them i.e. it will be implicitly called by the close(), BUT if you want to flush before closing any file you can use flush()

### **REMOVING WHITESPACES AFTER READING FROM TEXT FILE**

- read() and readline() reads data from file and return it in the form of string and readlines() returns data in the form of list.
- All these read function also read leading and trailing whitespaces, new line characters. If we want to remove these characters you can use functions
  - strip() : removes the given character from both ends.
  - lstrip(): removes given character from left end
  - .rstrip(): removes given character from right end

### **File Pointer**

- Every file maintains a file pointer which tells the current position in the file where reading and writing operation will take.
- When we perform any read/write operation two things happens:
  - The operation at the current position of file pointer
  - File pointer advances by the specified number of bytes.

## **1-MARK QUESTIONS**

1. To read three characters from a file object f, we use ..... .  
(a) f.read(3)            (b) f.read()            (c) f.readline()            (d) f.readlines()
2. The files that consists of human readable characters  
(a) binary file            (b) text file            (c) Both (a) and (b)            (d) None of these
3. Which function is used to write a list of string in a file?  
(a) writeline()            (b) writelines()            (c) writestatement()            (d) writefullline()
4. What will be the output of the following Python code?  
myFile = None  
for i in range (8):  
    with open("data.txt", "w") as myFile:  
        if i > 5:  
            break  
print(myFile.closed)
- (a) True            (b) False            (c) None            (d) Error
5. What will be the output of the following Python code?  
myFile = open("story.txt", "wb")  
print(" Name of the file: ", myFile.name)  
myFile.flush()  
myFile.close()

Suppose the content of the file "story.txt" is  
Education Hub

Learning is the key of success.

- (a) Compilation error    (b) Runtime error    (c) No output    (d) Flushes the file when closing them

6. What is the output of following code?

```
myfile=open('story.txt','r')
s=myfile.read(10)
print(s)
s1=myfile.read(15)
print(s1)
myfile.close()
```

- |                                    |                                    |                                       |                                    |
|------------------------------------|------------------------------------|---------------------------------------|------------------------------------|
| a) Education<br>Hub Learning<br>is | b) Education<br>Hub<br>Learning is | c) Education<br>Hub<br>Learning<br>is | d) Education<br>Hub<br>Learning is |
|------------------------------------|------------------------------------|---------------------------------------|------------------------------------|

7. What is the output of following code?

```
f=open('story.txt', 'r')
s=f.readline()
lcount=len(s)
print(lcount)
f.close()
```

- (a) 4            (b) 2            (c) 1            (d) 8

8. What is the output of following code?

```
f=open('story.txt', 'r')
str=""
s=0
ts=0
while str:
str=f.readline()
ts=ts+len(str)
print(ts)
f.close()
```

- (a) 44                      (b) 43                      (c) 37                      (d) 45

9. What is the output of following code?

```
def test () :
s=open ("story.txt", "r")
f=s.read()
z=f.split()
count=0
for i in z:
count=count+1
print(count)
```

Suppose the content of file ‘Para.txt’ is  
Education Hub  
Electronic learning  
Learning is the key of success.

- (a) 7                      (b) 8                      (c) 5                      (d) 9

10. What is the output of following code?

```
def test() :
f=open ("Para.txt", "r")
lines=0
l=f.readlines()
for i in l:
if i [0] == 'E' :
lines+=1
print(lines)
```

- (a) 2                      (b) 1                      (c) 3                      (d) Error

### **ANSWER – 1-MARK QUESTIONS**

1. (a) ) 2. (b) 3. (a) 4. (a) 5. (d) 6. (b) 7. (b) 8. (a) 9. (b) 10.

### **2 MARK QUESTIONS**

Q1. Write a single loop to display all the contents of a text file file1.txt after removing leading and trailing WHITESPACES

```
out=open('output.txt','w')
out.write('hello,world!\n')
out.write('how are you')
out.close()
open('output.txt').read()
```



Q3. Read the code given below and answer the questions

```
f1=open('main.txt','w')
f1.write('bye')
f1.close()
```

If the file contains 'GOOD' before execution, what will be the content of the file after execution of the code

Q4. Observe the following code and answer the follow

```
f1=open("mydata","a")
_____#blank1
f1.close()
```

(i) what type of file is mydata

(ii) Fill in the blank1 with statement to write "abc" in the file "mydata"

Q5. A given text file data.txt contains :

```
Line1\n
\n
line3
Line 4
\n
line6
```

What would be the output of following code?

```
f1=open('data.txt')
L=f1.readlines()
print(L[0])
print(L[2])
print(L[5])
print(L[1])
print(L[4])
print(L[3])
```

Q6. In which of the following file modes the existing data of the file will not be lost?

- i) rb
- ii) w
- iii) a+b
- iv) wb+
- v) r+
- vi) ab
- vii) w+b
- viii)wb
- ix) w+

Q7. What would be the data types of variables data in following statements?

- i) Data=f.read( )
- ii) Data=f.read(10)
- iii) Data=f.readline()
- iv)Data=f.readlines()

Q8 Suppose a file name test1.txt store alphabets in it then what is the output of the following code

```
f1=open("test1.txt")
size=len(f1.read())
print(f1.read(5))
```

Q9. What is standard input, output and error steams?

Q10. Write a short note on flush() function.

## ANSWER – 2-MARK QUESTIONS

Ans1 for line in open("file1.txt"):  
print(line.strip())

Ans2 The output will be  
Hello,world!  
How are you?

The first line of code is opening the file in write mode,the next two line writes text t file .the last line opens the file and from that reference reads the file content.file() performs the same functions as open().Thus,the file("output.txt")will give the references to open the file on which read() is applied.

Ans3 The file would now contains "Bye"only because when an existing file is openend in write mode .it truncates the existing data in file .

Ans4 i) Text file  
ii) f1.write("abc")

Ans5 Line1  
Line3  
Line 6  
Line 4

Ans6 ab and a+b mode

Ans7 a) string b)string c)string d)list

Ans 8 No Output

Explanation: the f1.read() of line 2 will read entire content of file and place the file pointer at the end of file. for f1.read(5) it will return nothing as there are no bytes to be read from EOF and, thus,print statement prints nothing.

**Ans 9**

- Standard input device(stdin) reads from the keyboard
- Standard output device(stdout)- prints to the display and can be redirected as standard input

Standard error device(stderr)- Same as stdout but normally only for errors. Having error output separately allows the user to divert regular output to a file and still be able to read error messages.

Ans 10

- While writing, Python writes everything into a buffer and pushes it into file at a later time.
- When flush() function is called it immediately empties the buffer and writes into the file.
- This function is called automatically when file is closed.

### 3 MARK QUESTIONS

**Q1. Write a python code to find the size of the file in bytes, number of lines and number of words.**

```
# reading data from a file and find size, lines, words
f=open('Lines.txt','r')
str=f.read( )
size=len(str)
print('size of file n bytes',size)
f.seek(0)
L=f.readlines( )
word=L.split( )
print('Number of lines ',len(L))
print('Number of words ',len(word))
f.close( )
```

**Q2. Write code to print just the last line of a text file “data.txt”.**

```
Ans: fin=open("data.txt","r")
lineList=fin.readlines()
print("Last line = ", lineList[-1])
```

**Q.3 Write a program to count the words “to” and “the” present in a text file “python.txt”.**

**Ans.**

```
fname = "python.txt"
num_words = 0
f= open(fname, 'r')
words = f.read().split()
    for a in words:
        if (a.lower() == "to" or a.lower() == "the" ):
            num_words = num_words + 1
print("Number of words:", num_words)
f.close()
```

**Q.4. Write a program to display all the lines in a file “python.txt” along with line/record number.**

**Ans.**

```
fh=open("python.txt","r")
count=0
lines=fh.readlines()
for a in lines:
    count=count+1
    print(count,a)
fh.close()
```

**Q. 5 Write a program to display all the lines in a file “python.txt” which have the word “to” in it.**

**Ans.**

```
fh=open("python.txt","r")
count=0
lines=fh.readlines()
for a in lines:
    if (a.lower().count("to") > 0) :
        print(a)
fh.close()
```

### **3-MARK QUESTIONS**

- 1. Write a python program to create and read the city.txt file in one go and print the contents on the output screen.**

**Answer:**

```
# Creating file with open() function
f=open("city.txt","w")
f.write("My city is very clean city.")
f.close()
# Reading contents from city.txt file
f=open("city.txt","r")
dt = f.read()
print(dt)
f.close()
```

- 2. Consider following lines for the file friends.txt and predict the output:**

```
Friends are crazy, Friends are naughty !
Friends are honest, Friends are best !
Friends are like keygen, friends are like license key !
We are nothing without friends, Life is not possible without friends !
```

```
f = open("friends.txt")
l = f.readline()
l2 = f.readline(18)
ch3=f.read(10)
print(l2)
print(ch3)
print(f.readline())
f.close()
```

**Output:**

```
Friends are honest
, Friends
are best !
```

**Explanation:**

In line no. 2, f.readline() function reads first line and stores the output string in l but not printed in the code, then it moves the pointer to next line in the file. In next statement we have f.readline(18) which reads next 18 characters and place the cursor at the next position i.e. comma (,) , in next statement f.read(10) reads next 10 characters and stores in ch3 variable and then cursor moves to the next position and at last f.readline() function print() the entire line.

- 3. Write a function count\_lines() to count and display the total number of lines from the file. Consider above file – friends.txt.**

```
def count_lines():
    f = open("friends.txt")
    cnt =0
    for lines in f:
        cnt+=1
        lines = f.readline()
    print("no. of lines:",cnt)
    f.close()
```

- 4. Write a function display\_oddLines() to display odd number lines from the text file. Consider above file – friends.txt.**

```
def display_oddLines():
    f = open("friends.txt")
    cnt =0
    for lines in f:
```

```

    cnt+=1
    lines = f.readline()
    if cnt%2!=0:
        print(lines)
    f.close()

```

**5. Write a function cust\_data() to ask user to enter their names and age to store data in customer.txt file.**

```

def cust_data():
    name = input("Enter customer name:")
    age=int(input("Enter customer age:"))
    data = str([name,age])
    f = open("customer.txt","w")
    f.write(data)
    f.close()

```

### 4 MARK QUESTIONS

Q1. This question consists of 6 sub-questions . Attempt any 5 questions.

Below is a program to delete the line having word (passed as argument). Answer the questions that follow to execute the program successfully.

```

def filedel(word) :
    file1 = open("Python.txt ","_____") # Statement 1
    nfile = open("algo.txt", "w")
    while True :
        line = file1._____ # Statement 2
        if not line :
            break
        else :
            if word in line : # Statement 3
                _____
            else : _____
                print(line)
                nfile. _____ (line) # Statement 4
        file1.close()
        _____ . close() # Statement 5
filedel('write')

```

(i). In which mode, program should open the file to delete the line in statement 2?

- (a) w                      (b) r                      (c) r+                      (d) a+

(ii). Choose the correct option to fill up the blank in line marked as Statement 3.

- (a) read()                      (b) read(n)                      (c) readlines()                      (d) readline()

(iii). Identify the missing code for blank space in line marked as Statement 4.

- (a) True                      (b) Flag                      (c) pass                      (d) False

(iv). Choose the correct option to fill up the blank in line marked as Statement 5.

- (a) read                      (b) write                      (c) writelines                      (d) writeline

(v). Choose the correct option to fill up the blank in line marked as Statement 6.

- (a) file 1                      (b) file                      (c) nfile                      (d) None

Answer – 1

- (i). (b)                      (ii). (d)                      (iii). (c)                      (iv). (b)                      (v). (c)

Q2. This question consists of 6 sub-questions . Attempt any 5 questions.

Below is a program to display all the records in a file along with line/record number from file "status.txt". Answer the questions that follow to execute the program successfully.

```

f= open (_____, "r")           #Line 1
count= 0
rec=""
while_____:                   #Line 2
    rec = f . _____      #Line 3
    if rec == " ":
                                #Line 4
        count = _____   #Line 5
print(count, rec, end = " ")
_____                       #Line 6

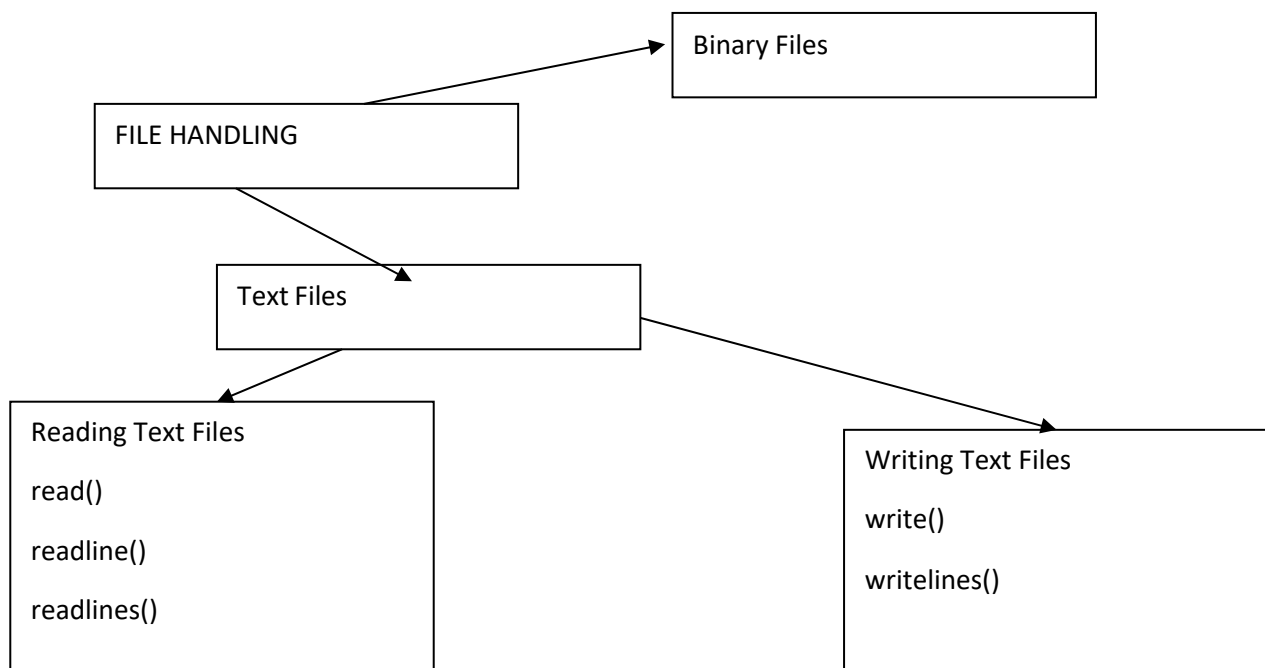
```

- (i). Choose the correct option to fill up the blank marked as Line 1.  
 (a) status      (b) "status.txt"      (c) status.txt      (d) file.txt
- (ii). Choose the correct option to fill up the blank marked as Line 2.  
 (a) 0      (b) 1      (c) False      (d) True
- (iii). Which function will be used to read the content of file marked as Line 3?  
 (a) readline()      (b) readlines()      (c) read()      (d) read(n)
- (iv). Choose the correct option to fill up the blank marked as Line 4.  
 (a) continue      (b) break      (c) goto      (d) label
- (v) Which value will be assign to variable count in Line 5?  
 (a) count -1      (b) count\*i      (c) count +1      (d) count + i
- (vi). Identify the missing code in Line 6.  
 (a) f.close      (b) myfile.close      (c) file.close()      (d) f.close()

Answer – 2

- (i). (b)      (ii). (d)      (iii). (a)      (iv). (b)      (v). (c)      (vi). (d)

### MIND MAP



## File Handling (Binary File)

There are two types of files:

**Text Files-** A file whose contents can be viewed using a text editor is called a text file. A text file is simply a sequence of ASCII or Unicode characters. Python programs, contents written in text editors are some of the example of text files.e.g. .txt,.rtf,.csv etc.

**Binary Files-**A binary file stores the data in the same way as as stored in the memory. The .exe files,mp3 file, image files, word documents are some of the examples of binary files.we can't read a binary file using a text editor.e.g. .bmp,.cdr etc.

Text File	Binary File
Its Bits represent character.	Its Bits represent a custom data.
Less prone to get corrupt as change reflects as soon as made and can be undone.	Can easily get corrupted, corrupt on even single bit change
Store only plain text in a file.	Can store different types of data (audio, text,image) in a single file.
Widely used file format and can be opened in any text editor.	Developed for an application and can be opened in that application only.
Mostly .txt,.rtf are used as extensions to text files.	Can have any application defined extension.

Opening and closing of binary file is same as text file opening and closing. While opening any binary file we have to specify 'b' in file opening mode.

Python has a module which does this work for us and is extremely easy to use. This module is called **pickle**; it provides us with the ability to serialize and deserialize objects, i.e., to convert objects into bitstreams which can be stored into files and later be used to reconstruct the original objects.

**pickle.dump()** function is used to store the object data to the file. It takes 3 arguments.First argument is the object that we want to store. The second argument is the file object we get by opening the desired file in write-binary (wb) mode. And the third argument is the key-value argument.

**Pickle.load()** function is used to retrieve pickled data.The steps are quite simple. We have to use pickle.load() function to do that.

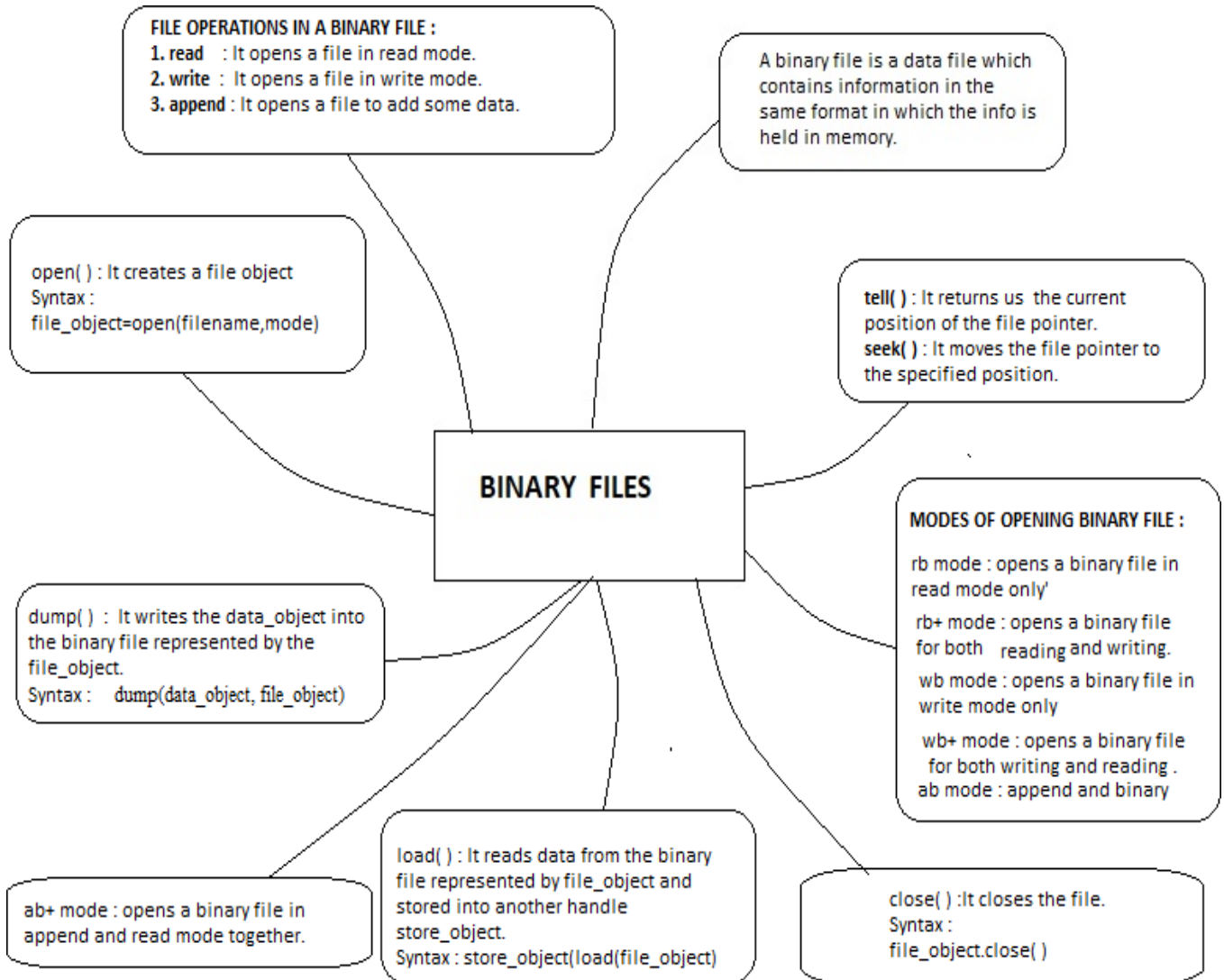
A Binary file is a file that contains information in the same format in which the information is held in memory, ie the file content that is returned to us is raw ( with no translation or specific encoding )

- File Handling consists of following three steps :
  - a) Open the file.
  - b) Process the file ie perform read or write operation.
  - c) Close the file.
- The **open( )** function is used to open a data file in a program through a file-object ( or a file handle).
- A file mode specifies the type of operation(s) ( eg read / write / append ) possible in the opened file ie it refers to how the file will be used, once it is opened.

- The **close( )** function breaks the link of the file-object and the file on the disk.
- A binary file can be opened in the modes ( 'rb', 'wb', 'ab', 'rb+', 'wb+', 'ab+')
- Pickle module in Python provides us, with the ability to **serialize** and **de-serialize** objects, ie to convert objects into bitstreams which can be stored into files and later be used to reconstruct the original objects.
- Serialization is also known as pickling.
- De-serialization is also known as un-pickling.
- Pickle module contains **dump( )** function to perform pickling ie write data into a binary file.
- Pickle module contains **load( )** function to perform unpickling ie read data from a binary file.
- Syntax of dump( ) function to write data :  
dump(data\_object, file\_object)  
Where data\_object is the object that has to be written(dumped) to the file with the file handle named file\_object
- Syntax of load( ) function to write data :  
store\_object= load(file\_object)  
The pickled Python object is read (loaded) from the file having the file handle named file\_object and is stored in a new file handle called store\_object.
- **Read from and Write into a binary file :**  
Use 'rb' mode in the open( ) function to read binary files.  
We hve **read( )** and **write( )** methods that works with string parameters and will not work directly with binary files. Conversion of data at the time of reading and writing is required.  
Dump( ) method is used to write the objects in a binary file and load( ) method is used to read data from a binary file.
- **Search operation :**  
There is no pre-defined function available in python for searching records. It involves reading of the binary file and then comparing each record with our given value. ( This is the linear search process )
- **Append operation :**  
To add/append data to a binary file, we must open a file in 'ab' mode. The file will retain the previous records and append the new records at the end of the file.
- **Update operation :**  
In order to update a binary file, the position of the file pointer must be known. To check the position of the file pointer, we use the functions **tell( )** and **seek( )**
- **tell( )** function : It returns the current position of the file pointer in the file.  
Syntax of tell( ) function : variable=file\_object.tell( )
- **seek( )** function : It changes the position of the file pointer by placing the file pointer at the specified position in the open file.  
Syntax of seek( ) function : file\_object.seek( offset)



## MIND MAP: Binary File



### 1 Mark Questions

1. Binary files commonly have the extension : 1  
a) .TXT      b) .DAT      c) .DOC      d) .PDF
2. The method\_\_\_\_\_is used to read data from a binary file. 1
3. State whether **True** or **False** : 1  
When we open a file in append mode the previous data is erased.
4. Which module is required to use the built-in function dump() ? 1  
a) math      b) flush      c) pickle      d) unpickle
5. The mode used for both writing and reading data from a binary file : 1  
a) wb+      b) w      c) wb      d) a
6. State whether **True** or **False** : 1  
The type of operation that can be performed on a file depends upon the mode in which the file is opened.
7. The module **pickle** is used for which of the methods in file handling : 1  
(a) open( ) and close( )  
(b) read( ) and write( )  
(c) dump( ) and load( )  
(d) None of the above
8. The\_\_\_\_\_statement automatically closes the file after processing on the file gets over. 1
9. The\_\_\_\_\_method is used to read data from a binary file. 1
10. The method used to write an object into a binary file is : 1  
a) load()      b) dump()      c) write()      d) None of these

### 2 Marks Questions

1. State whether **True** or **False** : 2  
a) Data stored inside the file is permanent in nature.  
b) A binary file can be opened in write mode using “w” mode only.
2. Write two features of Binary files . 2
3. Write a statement in Python to perform the following operations : 2  
a) To open a Binary file “BOOK.DAT” in read mode  
b) To open a Binary file in read and write mode
4. Observe the following code and answer the questions that follow : 2  
File = open(“Mydata”,“a”)  
\_\_\_\_\_ # Blank 1  
File.close()  
  
a) What type of file( Text / Binary ) is Mydata ?  
b) Fill the Blank 1 with statement to write “ABC” in the file Mydata
5. Q (i) and (ii) are ASSERTION ( A ) and REASONING ( R ) based questions. 2  
Mark the correct choice as :  
(A) Both A and R are true and R is the correct explanation for A.  
(B) Both A and R are true and R is not correct explanation for A.  
(C) A is true but R is false.  
(D) A is false but R is true.

Q (i)

Assertion ( A ) : Access mode 'a' adds data to the file.

Reason ( R ) : If file exists data added to end of file, if file does not exists a new file is created.

Q(ii)

Assertion ( A ) : Access mode 'a' opens a file for appending.

Reason ( R ) : The file pointer is at the end of the file if the file exists.

6. Write in short : 2
- i) What is the use of open( ) method ?
  - ii) What is the use of tell( ) method ?
7. What is the use of load( ) and dump( ) methods? 2
8. Answer in brief : 2
- i) Purpose of close( ) function.
  - ii) Purpose of flush( ) function.
9. Explain seek( ) method. 2

### 3 Marks Questions

1. a) Explain the wb+ mode in binary files. 3  
b) Explain the ab+ mode in binary files.  
c) Explain the rb mode in binary files.
2. a) What do you mean by serialization and de-serialization ? 3  
b) What are the usage of dump( ) and load( ) functions ?  
c) What are the usage of close( ) and flush( ) functions ?
3. Write a program that reads a binary file "emp.dat" and displays all records of employees one by one. 3
4. Write a program that reads a binary file "emp.dat" and display the records of all those employees who are getting salary greater than 20000. 3
5. a) Which method is used to write list, tuple and sequence data types in a binary file ? 3  
b) Which function forces Python to write the contents of buffer to file?  
c) Which area is automatically associated with the file when we open it?

### 4 Marks Questions

1. Consider a binary file EMP.DAT containing details empno:name:salary : 4  
(separator ':') Write a function Readfile( ) to display details of those employees who are earning between 20000 and 40000 ( both values inclusive)
2. For a Binary file, explain the processes : 4
- i) Search operation
  - ii) Append operation
  - iii) Update operation

iv) tell(0 function

3. Create the file phonebook.dat that stores the details in following format : 4  
Name Phone  
Alex 7865443  
Neha 8855342  
Obtain the details from the user.

4. A file phonebook.dat stores the details in following format : 4  
Name Phone  
Alex 7865443  
Neha 8855342  
Write a program to edit the phone numbers of "Anish" in file. If there is no record for "Anish" report error.

### 5 Marks Questions

1. Write a function to write numbers into a binary file and read the same. 5
2. Read the following passage and answer the questions that follow : 5

Ramesh, was assigned an incomplete task search( ) function whose purpose was to search in a pickled file student.dat The following information about student.dat is known :

- File contains details of students in [ roll, name, marks ] format
- File contains details of 10 students ( ie from roll 1 to 10 ) ans separate list of each student is written in the binary file using dump( )

Ramesh has been assigned the task to complete the code and print details of roll number 1 .

```
def search( ) :  
    f=open("student.dat",_____) # statement-1  
    _____: # statement-2  
  
    while True :  
        rec=pickle._____ # statement-3  
        if(_____) : # statement-4  
            print(rec)  
    except : pass  
        _____ # statement-5
```

- i) In which mode Ramesh should open the file in statement-1 ?  
a) r b) r+ c) rb d) wb
- ii) Identify the suitable code to be used at blank space in statement-2.  
a) if(rec[0]==1) b) for i in range (10) c) try d) pass

- iii) Identify the function (with argument) to be used at blank space in statement-3.  
a) load( )    b) load(student.dat)    c) load(f)    d) load(fin)
- iv) What will be the suitable code for blank space in statement-4?  
a) rec[0]==2    b) rec[1]==2    c) rec[2]=2    d) rec[0]==1
- v) Which statement should Ramesh use at blank space in statement-5 to close the file?  
a) file.close( )    b) close(file)    c) f.close( )    d) close( )

3. A binary file "Book.dat" has structure [BookNo, Book\_Name, Author, Price].

1. Write a user defined function CreateFile() to input data for a record and add to Book.dat .
2. Write a function CountRec(Author) in Python which accepts the Author name as parameter and count and return number of books by the given Author are stored in the binary file "Book.dat
- 3.

4. A binary file "STUDENT.DAT" has structure [admission\_number, Name, Percentage]. Write a function countrec() in Python that would read contents of the file "STUDENT.DAT" and display the details of those students whose percentage is above 75. Also display number of students scoring above 75%.

5 Write a function in python to search and display details, whose destination is "Cochin" from binary file "Bus.Dat". Assuming the binary file is containing the following elements in the list:

1. Bus Number
2. Bus Starting Point
3. Bus Destination
- 4.

6. Write a function addrec() in Python to add more new records at the bottom of a binary file "STUDENT.dat", assuming the binary file is containing the following structure :  
[Roll Number, Student Name]

7. Write a function searchprod( pc) in python to display the record of a particular product from a file product.dat whose code is passed as an argument. Structure of product contains the following elements [product code , product price]

8. Write a function routechange(route number) which takes the Route number as parameter and modify the route name(Accept it from the user) of passed route number in a binary file "route.dat".

9. Write a function countrec(sport name) in Python which accepts the name of sport as parameter and count and display the coach name of a sport which is passed as argument from the binary file "sport.dat". Structure of record in a file is given below —————  
[sport name, coach name]

10. A binary file "salary.DAT" has structure [employee id, employee name, salary]. Write a function countrec() in Python that would read contents of the file "salary.DAT" and display the details of those employee whose salary is above 20000.

11. Amit is a monitor of class XII-A and he stored the record of all the students of his class in a file named "class.dat". Structure of record is [roll number, name, percentage]. His computer teacher has assigned the following duty to Amit  
Write a function remcount( ) to count the number of students who need remedial class (student who scored less than 40 percent)

## Marking Scheme

1 Mark Questions		
1.	b) .DAT	1
2.	load()	1
3.	False	1
4.	c) pickle	1
5.	a) wb+	1
6.	True	1
7.	(c) dump( ) and load( )	1
8.	with	1
9.	load()	1
10.	b) dump()	1
2 Marks Questions		
1.	a) True b) False	2
2.	i) It can handle large files. ii) It consists of data with a specific pattern without any delimiter,	2
3.	a) f1=open("BOOK.DAT","r") b) f2=open("BOOK.DAT","w+")	2
4.	a) Textfile. b) File.write("ABC")	2
5.	i) Option B is correct. ii) Option A is correct.	2
6.	i) It opens a file on disk and associates it with a file handle. ii) It tells us the current position of the file pointer within the file.	2
7.	load( ) function is used for reading data fromn a binary file. dump( ) function is to write the object in a binary file.	2
8.	i) It breaks the link of the file object and the file on disk. After file is closed, no file operations can be don on the file.  ii) It forces the writing of data on the disk still pending in output buffers.	2
9.	Syntax is seek( offset, from_what) It changes the current file pointer position. The offset argument indicates the number of bytes to be moved. The from_what argument specifiesthe reference [position from where the bytes are to be moved.	2
3 Marks Questions		

1.	<p>a) wb+ mode is used to open a binary file in write and read modes both.</p> <p>b) ab+ mode is used to open a binary file in append and read mode both. Previous content is preserved.</p> <p>c) rb mode is used to open a binary file in read only mode. No other operation can be performed.</p>	3
2.	<p>a) Serialization is to convert objects into bitstreams which can be stored into files, whereas de-serialization is used to reconstruct the original objects. It is also known as pickling.</p> <p>b) The dump( ) function is used to perform pickling ie write data into a binary file. Whereas the load( ) function is used to perform unpickling ie read data from a binary file.</p> <p>c) close( ) function delinks the file_object with the file on the disk. The flush( ) function forces the write operation on the disk file which remains in the pending status.</p>	3
3.	<pre>import pickle f1=open("emp.dat","rb") e=pickle.load(f1) for x in e :     print(x) f1.close( )</pre>	3
4.	<pre>import pickle f1=open("emp.dat","rb") e=pickle.load(f1) for x in e :     if(e[x]&gt;20000) :         print(x) f1.close()</pre>	3
5.	<p>a) writelines( )</p> <p>b) flush( )</p> <p>c) buffer</p>	3
<b>4 Marks Questions</b>		
1.	<pre>def Readfile( ) :     i=open("EMP.DAT","rb+")     x=i.readline()     while(x) :         l=x.split(':')         If ((float(l[2])&gt;=20000) and ( float(l[2])&lt;40000)) :             print(x)         x=i.readline()     i.close()</pre>	4
2.	<p><b>i) Search operation :</b> There is no pre-defined function available in python for searching records. It involves reading of the binary file and then comparing each record with our given value. ( This is the linear search process )</p> <p><b>ii) Append operation :</b></p>	4

	<p>To add/append data to a binary file, we must open a file in 'ab' mode. The file will retain the previous records and append the new records at the end of the file.</p> <p><b>iii)</b> Update operation : In order to update a binary file, the position of the file pointer must be known. To check the position of the file pointer, we use the functions <i>tell()</i> and <i>seek()</i></p> <p><b>iv)</b> <i>tell()</i> function : It returns the current position of the file pointer in the file.</p> <p>Syntax of <i>tell()</i> function : <code>variable=file_object.tell()</code></p>	
3.	<pre>fp1=open("phonebook.dat","w") fp1.write("Name") fp1.write(" ") fp1.write("Phone") fp1.write("\n") while True :     name=input("Enter name : ")     phno=input("Enter phone no : ")     fp1.write(name)     fp1.write(" ")     fp1.write(phno)     fp1.write("\n")     ch=input("Want to enter more = y / n :")     if ch=='N' or ch=='n' :         break fp1.close()</pre>	4
4.	<pre>fp=open("phonebook.dat","w+") list="" while list :     pos=fp.tell()     list=fp.readline()     name,phone=list.split()     if name=="Anish" :         phone=input("Enter new number : ")         fp.seek(pos,0)         fp.write(name)         fp.write(" ")         fp.write(phone)     fp.close()     break</pre>	4
<b>5 Marks Questions</b>		
1.	<pre>def binfile() :     import pickle     file=open("mydata.dat","wb")     while True :         x=int(raw_input("Enter number : "))         pickle.dump(x,file)</pre>	5



	<pre> ans=raw_input("More data y / n : ") if ans.upper() == 'N' :     break file.close() </pre>	
2.	<p>i) Correct option is c) rb  ii) Correct option is c) try  iii) Correct option is c) load(f)  iv) Correct option is d) rec[0]==1  v) Correct option is c) f.close( )</p>	5
3	<pre> import pickle def createfile():     fobj=open("Book.dat","ab")     BookNo=int(input("Enter Book Number : "))     Book_name=input("Enter book Name :")     Author = input("Enter Author name: ")     Price = int(input("Price of book : "))     rec=[BookNo, Book_name ,Author, Price]     pickle.dump(rec, fobj)     fobj.close()  createfile() # This function is called just to verify result and not required in exam  def countrec(Author):     fobj=open("Book.dat", "rb")     num = 0     try:         while True:             rec=pickle.load(fobj)             if Author==rec[2]:                 num = num + 1                 print(rec[0],rec[1],rec[2],rec[3])     except:         fobj.close()     return num  n=countrec("amit") # This function is called just to verify result and not required in exam print("Total records", n) # This statement is just to verify result and not required in exam </pre>	5
4	<pre> import pickle def countrec():     fobj=open("student.dat","rb")     num = 0     try:         while True:             rec=pickle.load(fobj)             if rec[2]&gt;75: </pre>	5

	<pre> num = num + 1 print(rec[0],rec[1],rec[2]) except:     fobj.close() return num </pre>	
5	<pre> import pickle def countrec():     fobj=open("bus.dat","rb")     num = 0     try:         while True:             rec=pickle.load(fobj)             if rec[2]=="Cochin" or rec[2]=="cochin":                 num = num + 1                 print(rec[0],rec[1],rec[2])     except:         fobj.close()     return num n=countrec() # This function is called just to verify result print(n) </pre>	5
6	<pre> import pickle def addrec():     fobj=open("student.dat","ab")     rollno=int(input("Roll Number : "))     sname=input("Student Name :")     rec=[rollno,sname]     pickle.dump(rec,fobj) fobj.close() addrec() </pre>	3
7	<pre> import pickle def searchprod(pc):     fobj=open("product.dat","rb")     num = 0     try:         while True:             rec=pickle.load(fobj)             if rec[0]==pc:                 print(rec)     except:         fobj.close() n=searchprod(1) # This function is called to verify the result </pre>	5
8	<pre> import pickle def routechange(rno):     fobj=open("route.dat","rb")     try:         while True:             rec=pickle.load(fobj)             if rec[0]==rno:                 rn=input("Enter route name to be changed ") </pre>	5

	<pre> rec[1]=rn print(rec) #This statement is called to verify the change in the record except: fobj.close()  routechange(1) # This function is called to verify the result </pre>	
9	<pre> def countrec(sn): num=0 fobj=open("data.dat","rb") try: print("Sport Name","\t","Coach Name") while True: rec=pickle.load(fobj) if rec[0]==sn:  print(rec[0,"\t\t",rec[1]) num=num+1 return num except: fobj.close() </pre>	5
10	<pre> def countrec(): num=0 fobj=open("data.dat","rb") try: print("Emp id\tEmp Name\tEmp Sal") while True: rec=pickle.load(fobj) if rec[2]&gt;20000: print(rec[0,"\t\t",rec[1],"\t\t",rec[2]) except: fobj.close() countrec()# This function is called to verify the result </pre>	5
11	<pre> def countrec(): fobj=open("data.dat","rb") try: print("Emp id\tEmp Name\tEmp Sal") while True: rec=pickle.load(fobj) if rec[2]&gt;20000: print(rec[0,"\t\t",rec[1],"\t\t",rec[2]) except: fobj.close() countrec()# This function is called to verify the result </pre>	5

# File Handling (CSV File)

**CSV (Comma Separated Values)** is a file format for data storage which looks like a text file. The information is organized with one record on each line and each field is separated by comma.

## CSV File Characteristics

- One line for each record
- Comma separated fields
- Space-characters adjacent to commas are ignored
- Fields with in-built commas are separated by double quote characters.

## When Use CSV?

- When data has a strict tabular structure
- To transfer large database between programs
- To import and export data to office applications, Qedoc modules
- To store, manage and modify shopping cart catalogue

## CSV Advantages

- CSV is faster to handle.
- CSV is smaller in size
- CSV is easy to generate
- CSV is human readable and easy to edit manually
- CSV is simple to implement and parse
- CSV is processed by almost all existing applications

## CSV Disadvantages

- No standard way to represent binary data
- There is no distinction between text and numeric value.
- Poor support of special characters and control characters.
- CSV allows to move most basic data only. Complex configurations cannot be imported and exported this way
- Problem.s with importing CSV into SQL (no distinction between NULL and quotes)

## Write / Read CSV FILE

Writing and reading operation from text file is very easy. First of all we have to import csv module for file operation/method call.

**Writing:** we open file in 'w' writing mode using open() method which create newFile like object. Through csv.writer() method ,we create writer object to call writerow() method to write objects.

**Reading:** we open the file in 'r' mode and create newFile like object,further we create newfilereader object using csv.reader()method to read each row of the file.

## CSV File

- CSV (Comma-separated values) is a common data exchange format used by the applications to produce and consume data.
  - A CSV file is a simple text file where each line contains a list of values (or fields) delimited by commas (mostly), but you will encounter CSV files where data is delimited using tab (\t) or pipe (|) or any other character.
  - The first line of the CSV file represents the header containing a list of column names in the file.
  - CSV file is commonly used to represent tabular data
- See the following table

Name	Class	Section
Amit	X	A
Sumit	XI	B
Ashish	XII	C

The above table will be stored in CSV format as follows:

```
Name ,Class,Section
Amit,X,A
Sumit,XI,B
Ashish,XII,C
```

If the values in the table contain comma(,) like below in column Address

Name	Class	Address
Amit	X	Fr. Agnel School, Delhi
Sumit	XI	Fr. Agnel School, Delhi

Then in CSV it will be stored like below (Values containing comma will be enclosed in double quotes)

```
Name,Class ,Address
Amit,X,"Fr. Agnel School, Noida"
Sumit,XI,"Fr. Agnel School, Noida"
```

### Python Provides CSV module to work with csv file:

Main Functions are:

1. reader()
2. writer()
3. DictReader()
4. DictWriter()

1. **reader() function :** This function help us to read the csv file. This function takes a file object and returns a `_csv.reader` object that can be used to iterate over the contents of a CSV file.

### How to read entire data from data.csv file?

Let we try to read the following file ie “data.csv”

data.csv

CODING	OUTPUT
<pre>import csv f = open("data.csv", 'r') row = csv.reader(f) for i in row :     print(i) f.close()</pre>	<pre>['Name', 'Class', 'Subject'] ['Amit', 'XII', 'CS'] ['Sumit', 'X', 'IP']</pre> <p>Notice that each line in the CSV file is returned as a list of strings</p>

Line-wise explanation of the code

1. The first line is importing a csv module.
2. This line is opening a file (data.csv) in reading mode and f is file object or file handle.
3. This line is returning a csv.reader object that can be used to iterate over csv file contents
4. We are using loop on the object which is returned by the previous line.

CODING	OUTPUT
<pre>import csv f = open("data.csv", 'r') row = csv.reader(f) print(row) f.close()</pre>	Returns the memory address where csv.reader object is stored

### How to read specific column from data.csv?

CODING	OUTPUT	Actual Data of File
<pre>import csv f = open("data.csv", 'r') row = csv.reader(f) for i in row :     print(i[0] , i[2]) f.close() #The above code is reading first and third column from "data.csv" file</pre>	<pre>Name Subject Amit CS Sumit IP</pre>	<pre>['Name', 'Class', 'Subject'] ['Amit', 'XII', 'CS'] ['Sumit', 'X', 'IP']</pre>

If you want to skip the First Row (header of the table)

CODING	OUTPUT	Actual Data of File
<pre>import csv f = open("data.csv", 'r') row = csv.reader(f) next(row) for i in row :     print(i[0] , i[2]) f.close()  next () function will jump to the next row.</pre>	<pre>[ 'Amit' , 'CS' ] [ 'Sumit' , 'IP' ]</pre>	<pre>['Name', 'Class', 'Subject'] ['Amit', 'XII', 'CS'] ['Sumit', 'X', 'IP']</pre>
<pre>import csv f = open("data.csv", 'r') row = csv.reader(f) next(row) next(row) for i in row :     print(i[0] , i[2]) f.close() If we use the next() function two times then it will skip two lines from beginning</pre>	<pre>[ 'Sumit' , 'IP' ]</pre>	<pre>['Name', 'Class', 'Subject'] ['Amit', 'XII', 'CS'] ['Sumit', 'X', 'IP']</pre>

If there is any other delimiter other than comma like in the following file

```
id|name|email|salary
1|Amit|amit@mail.com|25000
2|Sumit|sumit@mail.com|30000
3|Ashu|ashu@mail.com|40000
```

CODING	OUTPUT
<pre>import csv f = open("data.csv", 'r') row = csv.reader(f, delimiter = '1') for i in row :     print(i) f.close()</pre>	<pre>[‘id’ , ‘name’ , ‘email’ , ‘salary’] [‘1’ , ‘Amit’ , ‘amit@mail.com’ , ‘25000’] [‘2’ , ‘Sumit’ , ‘sumit@mail.com’ , ‘30000’] [‘3’ , ‘Ashu’ , ‘ashu@mail.com’ , ‘40000’]</pre>

2. **writer() function** : This function help us to write data in a csv file. It accepts the same argument as the reader() function but returns a writer object (i.e \_csv.writer)

**There are two main methods used in writing in csv file**

1. **writerow()**
2. **writerows()**

**using writerow():**

CODING	DATA.CSV
<pre>import csv f = open("data.csv", 'w') wr = csv.writer(f) wr.writerow(['Name' , 'Class']) wr.writerow(['Amit' , 'XII']) f.close()</pre>	<pre>Name , Class ‘Amit’ , ‘XII’</pre>

**using writerows():**

CODING	DATA.CSV
<pre>import csv fields = ['Name' , 'Class' , 'Subject'] rows = [[‘Amit’ , ‘XII’ , ‘CS’] , [‘Sumit’ , ‘X’ , ‘IP’] , [‘Ashu’ , ‘XI’ , ‘CS’]] f = open("data.csv" , ‘w’ , newline = ‘ ‘) wr = csv.writer(f) wr.writerow(fields) wr.writerows(rows) f.close()</pre>	<pre>Name, Class, Subject Amit, XII, CS Sumit, X, IP Ashu, XI, CS</pre>

**Reading a CSV file with DictReader:**

This function(DictReader) is working similar to reader(). This function return lines as a dictionary instead of list.

CODING	OUTPUT	data.csv
<pre>import csv f = open("data.csv" , ‘r’) row = csv.DictReader(f) for i in row:     print(i)</pre>	<pre>{‘Name’ : ‘Amit’ , ‘Class’ : ‘XII’} {‘Name’ : ‘Sumit’ , ‘Class’ : ‘X’} {‘Name’ : ‘Ashu’ , ‘Class’ : ‘XI’}</pre>	<pre>Name, Class Amit, XII Sumit, X Ashu, XI</pre>

This CSV file has no header. So we have to provide field names via the fieldnames parameter

## CODING

```
import csv
f = open("data.csv", 'r')
row = csv.DictReader(f, fieldnames = ['Name', 'Class', 'Subject'])
for i in row:
    print(i)
f.close()
```

## OUTPUT

```
{'Name': 'Amit', 'Class': 'XII', 'Subject': 'CS'}
{'Name': 'Sumit', 'Class': 'X', 'Subject': 'IP'}
{'Name': 'Ashu', 'Class': 'XI', 'Subject': 'CS'}
```

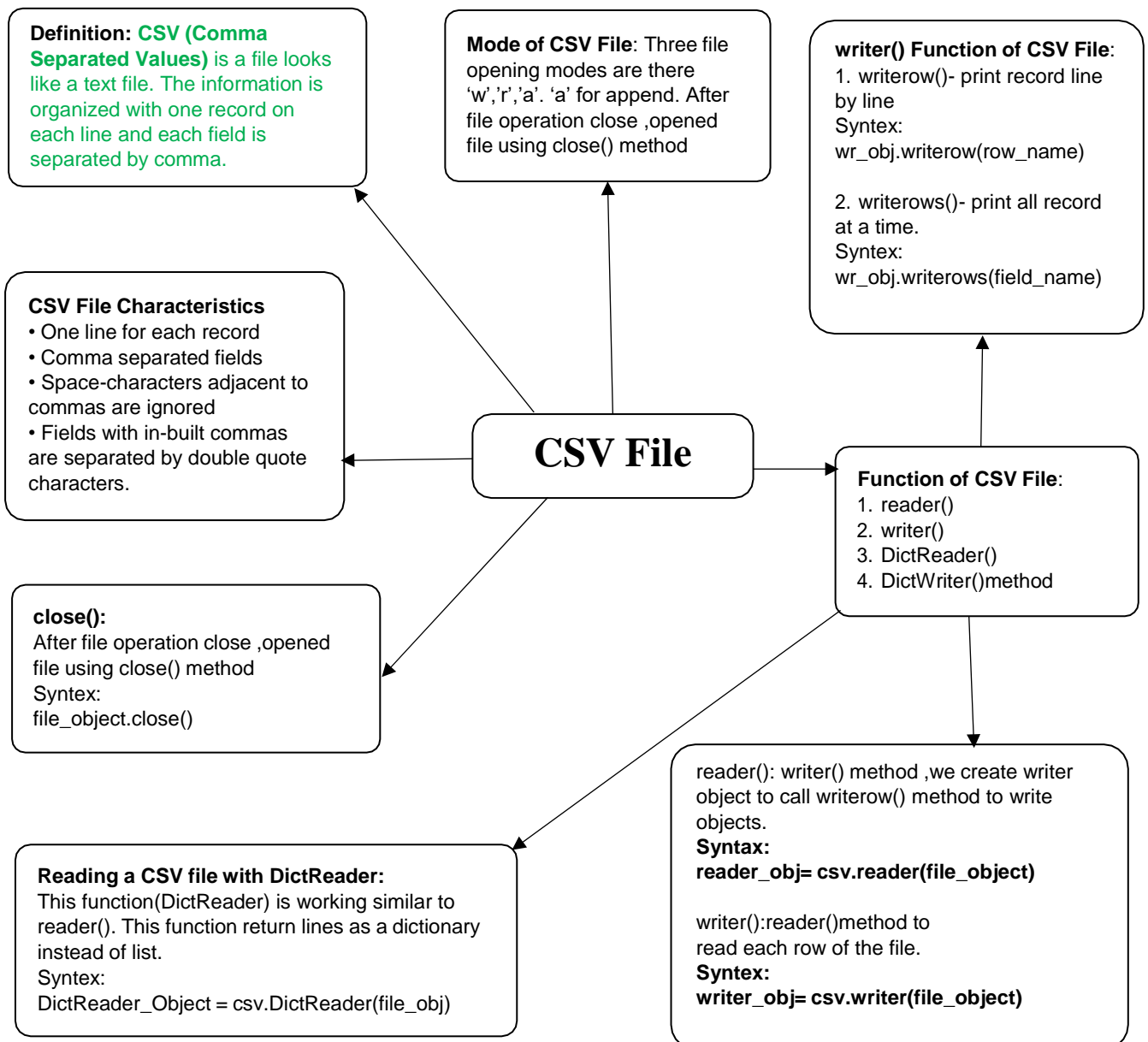
data.csv

Amit, XII, CS

Sumit, X, IP

Ashu, XI, CS

## Mind Map- CSV File





### 1 Marks

**Q1.** Which of the following module is provided by Python to do several operations on the CSV files?

- (A) os            (B) xls            (C) csv            (D) py

**Q2.** Which of the following is tasks cannot be done or difficult with CSV module?

- (A) Storing data in tabular form            (B) Finding data uniquely  
(C) Saving data permanently            (D) All of these

**Q3.** The writer() function of csv module has how many mandatory parameters?

- (A) 4            (B) 3            (C) 2            (D) 1

**Q4.** Which of these is not relevant to csv file?

- (A) Text file            (B) Binary file  
(C) Easily openable in text editor            (D) None of the above

**Q5.** You are having myfile.csv file on c drive. You want to access it in python, how can you do this?

- (A) source="c:\\myfile.csv" data = pd.read\_csv(source)  
(B) source="c:\\myfile.csv" data = pd.reads\_csv(source)  
(C) source="c:\\myfile.csv" data = pd.reader(source)  
(D) None of the above

### 2 Marks

**Q4.**

**Assertion:** To specify a different delimiter while writing into a csv file, *delimiter* argument is used with *csv.writer()*.

**Reason:** The CSV file can only take a comma as a delimiter.

- (A) Both Assertion and reason are true and reason is the correct explanation of assertion.  
(B) Assertion and reason both are true but reason is not the correct explanation of assertion.  
(C) Assertion is true, reason is false.  
(D) Assertion is false, reason is true.

**Q7.**

**Assertion:** Every record in a CSV file is stored in reader object in the form of a list.

**Reason:** writerow() method allows us to write a list of fields to the CSV file.

- (A) Both Assertion and reason are true and reason is correct explanation of assertion.  
(B) Assertion and reason both are true but reason is not the correct explanation of assertion.  
(C) Assertion is true, reason is false.  
(D) Assertion is false, reason is true.

**Q10.**

**Assertion:** The CSV module in Python's standard library presents methods to perform read/write operations on CSV files.

**Reasoning:** To write data to a CSV file, we can use built-in function *writer()*.

- (A) Both Assertion and reason are true and reason is the correct explanation of assertion.  
(B) Assertion and reason both are true but reason is not the correct explanation of assertion.  
(C) Assertion is true, reason is false.  
(D) Assertion is false, reason is true

**Q11.**

**Assertion:** To open a CSV file *employee.csv* in write mode, we can give python statement as follows:

```
fh=open('employee.csv')
```

**Reasoning:** If we try to write on a csv file that does not exist, the file gets Created

- (A) Both Assertion and reason are true and reason is the correct explanation of assertion.  
(B) Assertion and reason both are true but reason is not the correct explanation of assertion.  
(C) Assertion is true, reason is false.  
(D) Assertion is false, reason is true.

## 5 Marks

- Q1 Rohit, a student of class 12th, is learning CSV File Module in Python. During examination, he has been assigned an incomplete python code (shown below) to create a CSV File 'Student.csv' (content shown below). Help him in completing the code which creates the desired CSV File.

### CSV File

1,AKSHAY,XII,A  
2,ABHISHEK,XII,A  
3,ARVIND,XII,A  
4,RAVI,XII,A  
5,ASHISH,XII,A

### Incomplete Code

```
import _____ #Statement-1
fh = open(_____, _____, newline='') #Statement-2
stuwriter = csv._____ #Statement-3
data = []
header = ['ROLL_NO', 'NAME', 'CLASS', 'SECTION']
data.append(header)
for i in range(5):
    roll_no = int(input("Enter Roll Number : "))
    name = input("Enter Name : ")
    Class = input("Enter Class : ")
    section = input("Enter Section : ")
    rec = [_____] #Statement-4
    data.append(rec)
    stuwriter._____ (data) #Statement-5
fh.close()
```

- Identify the suitable code for blank space in line marked as Statement-1.  
a) csv file      b) CSV      c) csv      d) Csv
- Identify the missing code for blank space in line marked as Statement-2?  
a) "School.csv","w"      b) "Student.csv","w"  
c) "Student.csv","r"      d) "School.csv","r"
- Choose the function name (with argument) that should be used in the blank space of line marked as Statement-3  
a) reader(fh)      b) reader(MyFile)      c) **writer(fh)**      d) writer(MyFile)
- Identify the suitable code for blank space in line marked as Statement-4.  
a) 'ROLL\_NO', 'NAME', 'CLASS', 'SECTION'      b) ROLL\_NO, NAME, CLASS, SECTION  
c) 'roll\_no','name','Class','section'      d) roll\_no,name,Class,sectionc) co.connect()
- Choose the function name that should be used in the blank space of line marked as Statement-5 to create the desired CSV File?  
a) dump()      b) load()      c) writerows()      d) writerow()

## Marking Scheme

### 1 Marks

- Q1. (C) csv  
Q2. (B) Finding data uniquely  
Q3. (D) 1  
Q4. (B) Binary file  
Q5. (A) source="c:\\myfile.csv" data = pd.read\_csv(source)

### 2 Marks

- Q4. (C) Assertion is true, reason is false.  
Q7. (B) Assertion and reason both are true but reason is not the correct explanation of assertion.

**Q10.** (C) Assertion is true, reason is false.

**Q11.** (D) Assertion is false, reason is true.

### 5 Marks

Q1	
i.	Correct Answer : c) csv
ii.	Correct Answer : b) "Student.csv", "w"
iii.	Correct Answer: c) writer(fh)
iv.	Correct Answer : d) roll_no,name,Class,section
v.	Correct Answer : c) writerows()

## STACK Data Structure

### Syllabus :

Data Structure: Stack, operations on stack (push & pop), implementation of stack using LIST

**Data Structure** is the method of organising data in the memory of the computer for efficient processing of Data.

Types of Data Structures :

1. List
2. Stack
3. Queue

We already have studied Lists in class 11 and its review in class 12. The topic STACK is in the syllabus of class 12 . ( Queue is not in syllabus)

**Stack:** A stack is an arrangement of a set of homogeneous elements one upon another and is based on the principle of Last – In – First – Out (LIFO).

It has two major operations PUSH & POP .

**PUSH :** This operation adds an element at the top of the stack. Then the added element becomes the top of the Stack.

If the STACK is already full and no element can be pushed into the STACK , it is known as OVERFLOW.

**POP :** This operation removes an element from the top of the stack.

Then the previous element from the deleted position becomes the top of the Stack.

If the STACK is empty it is known as UNDERFLOW.

**Implementation of Stack in Python:** We can implement Stack in python using List. Let us write a program in python to create a stack of integer and perform some Push and Pop operation according to the user choice.

#Program: To implement Stack Using List.

```
def PUSH(stk,ele):                                     #Function to push element in stack.
    stk.append(ele)

def POP(stk):                                          #Function to pop element from stack.
    if len(stk) == 0:
        print("Stack is empty!")
    else:
        ele = stk.pop()

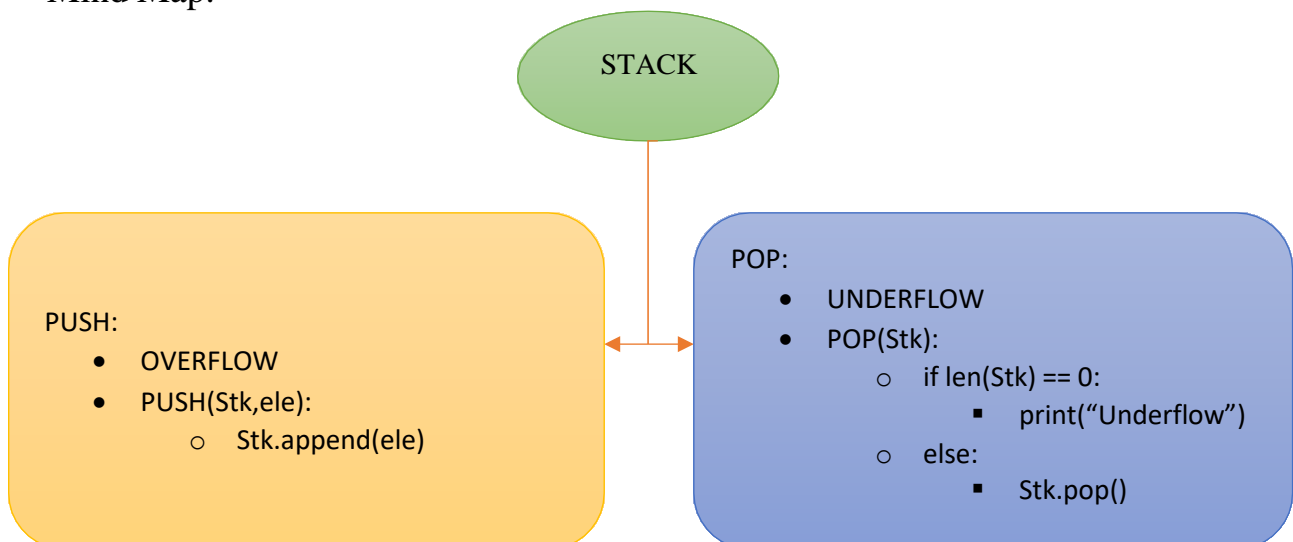
def DISPLAY(stk):                                     #Function to display elements of stack.
    L = len(stk)
    if L == 0:
        print("Stack is empty!")
```

```

else:
    print("The elements in the stacks are:")
    for i in range(L-1, -1, -1):
        print(stk[i])
#_main_____
stk = [] #Creating an empty List for Stack
while True:
    print ("PRESS 1 for Push \n PRESS 2 for Pop")
    print ("PRESS 3 for Display \n PRESS 4 for Exit")
    ch = int(input())
    if ch == 1:
        item = int(input("Enter data: "))
        PUSH(stk, item)
    elif ch == 2:
        POP(stk)
    elif ch == 3:
        DISPLAY(stk)
    elif ch == 4:
        break
    else:
        print ("Invalid Choice...")

```

Mind Map:



### APPLICATIONS OF STACK :

1. Reverse a word / line : In a Stack, the elements can be read from TOP end, ie the last element. Hence it is used to reverse a word or line.
2. The Compiler uses Stack to store the previous state of a program, when a function is called(activated) . ( During Recursion / Recursive funcrion calls)
3. Undo Mechanism in Text Editors.

**First of all , let us solve some basic questions :**

A) Fill in the blanks : ( These are 1 marks questions )

- i) means organization of data. (Ans: Data structure)
- ii) Stack is also known as .....list. (Ans: LIFO)
- iii) Insertion of an element in a Stack is known as ..... (Ans: push )
- iv) Deletion of an element from top of Stack is known as ..... (Ans: pop)

- v) LIFO stands for..... (Ans: Last-In-First-Out)
- B) State whether the statements are **True** or **False** : ( These are 1 marks questions )
- i) Insertion and Deletion from a Stack, takes place only from TOP. (Ans: True)
- ii) If a Stack is Empty, it is termed as STACK OVERFLOW. (Ans: False)
- iii) Removal of an element from top of stack is termed as POP. (Ans: True)
- iv) Undo mechanism of text editors is an application of Stack. (Ans: True)
- v) Insertion of an element to the top of stack is termed as PUSH. (Ans: True)

### Questions:

S.No.	Question	
Q 01	Stack is also known as: a) First-in First-out    b) Last-in-First-out    c) Both a and b    d) None of these	1
Ans	b) Last-in-First-out	
Q 02	In a stack, if a user tries to remove an element from empty stack, it is known as: a) Underflow    b) Empty    c) Overflow    d) None	1
Ans	a) Underflow	
Q 03	What type of data structure a stack is? a) Linear    b) Non-Linear    c) Both A and B    d) None	1
Ans	a) Linear	
Q 04	Which one of the following is an application of Stack Data Structure? a) Managing Function Call    b) Arithmetic Expression Evaluation c) Undo Redo    d) All of the above	1
Ans	d) All of the above	
Q 05	Which data structure we can use for reversing a string? a) Queue    b) List    c) Stack    d) Dictionary	1
Ans	c) Stack	
Q 06	Stack can be implemented in Python using: a) Tuple    b) Dictionary    c) Set    d) List	1
Ans	d) List	
Q 07	What are the two operations of stack?	1
Ans	The two operations of stack are Push and Pop	
Q 08	How do we test for empty stack?	1
Ans	if len(stk) == 0: print("Stack is Empty.")	
Q 09	Why do we use append() method of list in push operation?	1
Ans	To push data at the top of the stack.	
Q 10	ASSERTION AND REASONING based questions. Mark the correct choice as (a) Both A and R are true and R is the correct explanation for A (b) Both A and R are true and R is not the correct explanation for A (c) A is True but R is False (d) A is false but R is True Assertion (A): - In stack during pop operation it is necessary to check for underflow. Reasoning (R): - Underflow is the when the memory allocated to the stack gets full.	1
Ans	c) A is True but R is False	
Q 11	Define any two characteristics of stacks.	2

Ans	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 the first one to be out.	
<b>Q 12</b>	Define any two application of stacks.	2
Ans	Reversing a string. In text/image editor for redo/undo the editing.	
<b>Q 13</b>	Write down the status of Stack after each operation: Stack = [10,20,30,40] where TOP item is 40 Push 70 Push 100 Pop an item from Stack	2
Ans	<pre>  100    70     70     70     40     40     40     30     30     30     20     20     20     10  ,   10   and   10   </pre>	
<b>Q 14</b>	What will be the output of the following code: <pre> def PUSH(L):     Stk = []     for i in L:         if i % 5 == 0:             Stk.append(i)     for i in range(len(L)-1, -1, -1):         print(L[i])     else:         print("Stack is Empty") PUSH([12,23,34,54,56,67,87,89]) </pre>	2
Ans	Stack is Empty	
<b>Q 15</b>	What will be the output of the following code: <pre> def POP(Stk):     if len(Stk) == 0         print("Stack is Empty")     else:         while len(Stk) != 0:             x = Stk.pop()             if x[0] not in "AEIOUaeiou":                 print(x) Stk = ["Hello", "How are you", "I am fine", "Thank you"] POP(Stk) </pre>	2
Ans	Thank you How are you Hello	
<b>Q 16</b>	What will be the output of the following code: <pre> def G75(D):     Stk = []     for k in D:         if D[k] &gt; 75:             Stk.append(k) </pre>	2

	<pre> if len(Stk) == 0:     print("Stack is Empty") else:     while len(Stk) != 0:         print(Stk.pop()) D = {"Ajay":76, "Vijay":99, "Sanjay": 75} G75(D) </pre>	
Ans	Vijay Ajay	
<b>Q 17</b>	<p>Find the error in the following code and write the correct code and underline the corrected ones:</p> <pre> Def POP(Stk):     If len(Stk) == 0     print("Stack is Empty")     else     F = 1     while len(Stk) != 0:         F = F * Stk.pop()     return F </pre>	2
Ans	<pre> def POP(Stk):     if len(Stk) == 0     print("Stack is Empty")     else:     F = 1     while len(Stk) != 0:         F = F * Stk.pop()     return F </pre>	
<b>Q 18</b>	<p>Find the error in the following code and write the correct code and underline the corrected ones:</p> <pre> def PUSH(Stk, L): for w IN L:     if w[0] IN "AEIOUaeiou":         Stk.extend(w) </pre>	2
Ans	<pre> def PUSH(Stk, L):     for w in L:         if w[0] in "AEIOUaeiou":             Stk.append(w) </pre>	
<b>Q 19</b>	<p>Find the error in the following code and write the correct code and underline the corrected ones:</p> <pre> def Display(L)     l = len L     if l == 0:         print(Stack is Empty!!!)     else:         print("Elements in stack are:")         For i in range(l - 1, -1, -1):             print(L[i]) </pre>	2
Ans	<pre> def Display(L):     l = len(L) </pre>	

	<pre> if l == 0:     print("Stack is Empty!!!") else:     print("Elements in stack are:")     for i in range(l - 1, -1, -1):         print(L[i]) </pre>	
<b>Q 20</b>	Write a function in Python <b>PUSHD3(L)</b> , where L is a list of numbers. from this list push all numbers divisible by 3 into a stack implemented using a list. Display the stack if it has at least one element, otherwise display appropriate error message.	3
Ans	<pre> def PUSHD3(L):     Div3 = []     for i in L:         if i % 3 == 0:             Div3.append(i)     if len(Div3) == 0         print("Stack is Empty")     else:         while len(Div3) != 0:             print(Div3.pop()) </pre>	
<b>Q 21</b>	<p>Sakshi has created a dictionary D, containing names and salary as key value pairs of 5 employees. Write two separate functions to perform the following operations:</p> <ul style="list-style-type: none"> <li>• PUSH(HS, D), where HS is the stack and D is the dictionary which containing names and salaries. Push the keys (name of the employee) of the dictionary into a stack, where the corresponding value (salary) is greater than ₹75,000.</li> <li>• POP(HS), where HS is the stack. Pop and display the content of the stack.</li> </ul>	3
Ans	<pre> def PUSH(HS, D):     for k in D:         if D[k] &gt; 75000:             HS.append(k) def POP(HS):     if len(HS) == 0         print("Stack is Empty")     else:         while len(HS) != 0:             x = HS.pop()             print(x) </pre>	
<b>Q 22</b>	Write the functions in python, <b>PUSH(Stk, L)</b> and <b>POP(Stk)</b> , where Stk is the stack and L is a list of strings. In PUSH function read the list and push those string into the stack which start with a vowel. And in POP function display the stack.	3
Ans	<pre> def PUSH(Stk, L):     for w in L:         if w[0] in "AEIOUaeiou":             Stk.append(w) def POP(Stk):     if len(Stk) == 0         print("Stack is Empty")     else: </pre>	



	<pre> while len(Stk) != 0:     x = Stk.pop()     print(x) </pre>	
<b>Q 23</b>	<p>What will be the output of following code:</p> <pre> def PUSH(Stk, N)     for i in range(N,0,-1):         Stk.append(i) def POP(Stk):     if len(Stk) == 0         print("Stack is Empty")     else:         F = 1         while len(Stk) != 0:             F = F * Stk.pop()         return F Stk = [] PUSH(Stk, 5) print(POP(Stk)) </pre>	3
Ans	120	
<b>Q 24</b>	<p>Write a function DoubleStack(Stk), where Stk is a stack containing integers. In the function pop all the element of the stack and double the elements then push back to the stack in its original place.</p> <p>For example:  Stk = [10,20,30,40,50]  After the execution of the function the stack should be:  Stk = [20,40,60,80,100]</p>	3
Ans	<pre> Def DoubleStack(Stk):     tempStk = []     while len(Stk) != 0:         x = Stk.pop()         tempStk.append(x*2)     while len(tempStk) != 0:         x = tempStk.pop()         Stk.append(x) </pre>	
<b>Q 25</b>	<p>Write the functions in python, <b>PUSH(L, data)</b>, <b>POP(L)</b>, <b>DISPLAY(L)</b> to add a new integer, delete an integer from a List of numbers L and display the element of the stack, considering them to act as push and pop operations of the Stack data structure.</p>	4
Ans	<pre> def PUSH(L, data):     L.append(data) def POP(L):     if len(L) == 0:         print("Stack is empty!")     else:         return L.pop() def Display(L):     l = len(L)     if l == 0:         print("Stack is Empty!!!") </pre>	

	<pre> else:     print("Elements in stack are:")     for i in range(l - 1, -1, -1):         print(L[i]) </pre>	
<b>Q 26</b>	Write a program to reverse a string using stack.	4
Ans	<pre> def revStr(S):     stk = []     rS = ''     for i in S:         stk.append(i)     while len(S) != 0:         rS = rS + S.pop()     return rS S = input("Enter a String:") rS = revStr(S) print("Reverse String is:", rS) </pre>	
<b>Q 27</b>	Write a function PUSH(L) and POP(Stk), where L is List of integers and Stk is a stack. In PUSH function create two stack StkO and StkE, then read the list L and push all the even element in StkE and push all the odd element in StkO. At last return both the stacks. In POP function pop the number from the stack and display.	4
Ans	<pre> def PUSH(L):     StkO = []     StkE = []     for i in L:         if i%2 == 0:             StkE.append(i)         else:             StkO.append(i)     return StkO, StkE def POP(Stk):     if len(Stk) == 0         print("Stack is Empty")     else:         while len(Stk) != 0:             print(Stk.pop()) </pre>	
<b>Q 28</b>	<pre> def PUSH(Stack,N):     Stack._____( N)                #Statement 1 def POP(Stack):     if _____:                #Statement 2         return "Underflow"     else:         return Stack. _____ #Statement 3 Stack=[] Books=['Python', 'C', 'Java', 'C++', 'PHP', 'C#] for i in range(0,len(Books), _____): #Statement 4     PUSH(Stack,Books[i]) while len(Stack) != 0:     print(_____)                #Statement 5 </pre>	5

	<p>Required Output:  C  C++  C#  Fill the above statement based on given questions:</p>	
i	<p>Identify the suitable code for the blank of statement 1.  a) .append    b) .insert    c) .extend    d) .add</p>	
Ans	a) append	
ii	<p>Fill the statement 2, to check the stack is empty.  a) Stack = []                    b) Stack.isEmpty()  c) len(Stack) == 0    d) No of the above</p>	
Ans	c) len(Stack) == 0	
iii	<p>Fill the statement 3, to delete an element from the stack.  a) pop(1)                    b) pop()                    c) del Stack[1]                    d) delete(1)</p>	
Ans	b) pop()	
iv	<p>Fill the statement 4, to insert the alternate element from Books list.  a) 3    b) 0    c) -1    d) 2</p>	
Ans	d) 2	
v	<p>Fill the statement 5, to call the pop function.  a) pop(Books)                    b) pop(Stack)  c) call pop(Stack)                    d) def pop(Stack)</p>	
Ans	b) pop(Stack)	
<b>Q 29</b>	<pre> _____PUSH(S,N):                               #Statement 1     S.append(N) def POP(S):     if S != []:         _____S.pop()                       #Statement 2     else:         return None R={"OM":76, "JAI":45, "BOB":89, "ALI":65, "ANU":90, "TOM":82} ST=[] for k in R:     if R[k] &gt;= 75:         PUSH(_____)                               #Statement 3     _____True:                               #Statement 4     if ST != []:         print(POP(ST))     else:         _____                               #Statement 5 </pre> <p>Required Output:  TOM  ANU  BOB  OM  Fill the above statement based on given questions:</p>	5
i	<p>Fill the Statement 1, to define a function.  a) DEF                    b) Def                    c) def                    d) define</p>	
Ans	c) def	
ii	<p>Fill the Statement 2, to return the popped data.</p>	

	a) send      b) pass      c) assign      d) return	
Ans	d) return	
iii	Fill the Statement 3, to pass the proper arguments. a) ST, k      b) k, ST      c) ST      d) k	
Ans	a) ST, k	
iv	Fill the Statement 4, to complete the looping structure. a) for      b) do while      c) if      d) while	
Ans	d) while	
v	Fill the Statement 5, to stop the infinite loop. a) stop      b) Stop      c) break      d) Exit	
Ans	c) break	
<b>Q 30</b>	<pre> status =_____ #Statement 1 def PUSH(cust):     if cust[2] == "Goa":         L = [cust[0],cust[1]]         _____.append(L) #Statement 2 def POP():     if____(status) == 0: #Statement 3         print(_____) #Statement 4     while_____!= 0: #Statement 5         x = status.pop()         print(x) PUSH(["Gurdas", "9999999999","Goa"]) PUSH(["Julee", "8888888888","Mumbai"]) PUSH(["Murugan","7777777777","Cochin"]) PUSH(["Ashmit", "1010101010","Goa"]) POP() Required Output: ["Ashmit","1010101010"]                   ["Gurdas","9999999999"] </pre> <p>Fill the above statement based on given questions:</p>	5
i	Fill the Statement 1, to create a empty stack using list. a) ()      b) {}      c) []      d) ""	
Ans	c) []	
ii	Fill the Statement 2, to complete push operation. a) cust      b) status      c) cust[top]      d) status[top]	
Ans	b) status	
iii	Fill the Statement 3, to find the size of stack. a) size      b) length      c) count      d) len	
Ans	d) len	
iv	Fill the Statement 4, to display proper message where underflow occurs. a) "Stack is Empty"      b) Stack is Empty c) [Stack is Empty]      d) {Stack is Empty}	
Ans	a) "Stack is Empty"	
v	Fill the Statement 5, to iterate the loop properly. a) size(status)      b) len(status)      c) count(status)      d) length(status)	
Ans	len(status)	

# Networking

## Mind Map



## Introduction

Computer Network is a collection of autonomous computers interconnected by a single technology.

Two computers are said to be interconnected if they are able to exchange information. A computer network is a system that connects independent computers in order to share information and resources

### **Advantage of Computer Network:**

- Central Storage of Data
- Sharing of Information
- Sharing of Resources (Hardware & Software)
- Reliability
- Communication
- Reduced Cost

### **Disadvantage of Computer Network:**

- Computer networks require a specific setup
- Lack of Security
- Cost of network hardware and software

### **Components of Data Communication:**

- Message – It is information to be communicated
- Sender – The device which send the message
- Receiver – The device which receive the message
- Transmission media – It is physical path by which message travel from sender to receiver
- Protocol – It is set of rules that governs data communication. Actually, it is agreement between the sender and receiver regarding various communication parameter.

### **Data Flow**

- Simplex – In this mode of communication, data is transmitted in one direction only. e.g., Keyboard, monitor. It uses entire capacity of channel to send the data.
- Half Duplex – Communication is bi-directional but not at the same time. i.e., Walkie-Talkie. It uses entire capacity of channel is utilized for each direction.
- Full Duplex – Communications is bi-directional simultaneously i.e., both sender and receiver can send data at the same time.

### **Network Terminology**

- Node- The device connected to a network.
- Client – The device that requests for a service
- Server – The Device that renders the services
- Client-Server - In this model, the data are stored on powerful computers called **Server** that can be accessed by a much simpler computer called **Client** that are connected by a network.
- Network Interface Card or Unit (Network Adapter or LAN card) - It is hardware that allows a computer (or device) to connect to network.
- MAC (Media Access Control) address – Each NIC is assigned a unique 12-digit hexadecimal number, known a MAC address, is used as network address in communication. The format for the MAC address is  
MM : MM : MM : SS : SS : SS

Manufacturer ID      Card Id

- IP Address: Every device on network has unique identifier called IP address. It consists of 4 bytes (IPv4) decimal number (between 0 to 255) separated by ‘.’ (Period).
- Channel – It is communication path through which data is actually transmitted.

- Communication Media- It allows data or signal to be communicated across the devices. It is means of communication.
- Data – Information stored within the computer system in form of ‘0’ and ‘1’
- Signal- It is electric or electromagnetic encoding of data to be transmitted. It can be categorized into following two types:
  - Analog Signal – that has infinitely many levels of intensity over a period of time.
  - Digital Signal – that can have only a limited number of defined values.
- Bit rate – It defines the amount of data transferred. It is defined as number of bits per second (bps). [Bps – Bytes per Second]
- Baud – The number of changes in signal per second.
- Bandwidth – It is difference between the highest and the lowest frequencies contained in the signal.

### IP Address vs MAC Address

IP Address	MAC Address
It is of 4 bytes	It is of 6 bytes
Represented by decimal number	Represented by hexadecimal number
It is logical address	It is physical address
Its value can vary for the same machine	It is fixed address
It can be assigned only when a device is connected to network	It is assigned by manufacturer of the card irrespective of connectivity
Command to know the IP address is <b>Ipconfig</b>	Command to know the IP address is <b>ipconfig/all</b>

### Switching Technique

- A switched network consists of a series of interlinked nodes called switches capable of creating temporary connections between two or more liked devices.
- There are three basic switching technique
  - **Circuit Switching:** In circuit switching a dedicated path is established before sending data from sender to receiver and entire communication is carried out the same path.
  - **Packet Switching** – In packet switching in a message is broken into a number of parts called packet which are sent independently from sender to receiver and reassembled at the destination.
  - **Message Switching:** Message switching is a connectionless network switching technique where the entire message is routed from the source node to the destination node, one hop at a time. It was a precursor of packet switching.

### Circuit Switching vs Packet Switching

Circuit Switching	Packet Switching
A dedicated path is established	No dedicated path is established
Entire message follows the same path	Each packet travels independently to each other

### Network Devices

- **Modem**
  - It stands for modulator and demodulator
  - It a computer hardware device that converts data from a digital format into a format suitable for an analog.

- A modem transmits data by modulating one or more carrier wave signals to encode digital information, while the receiver demodulates the signal to recreate the original digital information.
- **Repeater**
  - Repeaters are network devices that amplify or regenerate an incoming signal before retransmitting it.
  - It operates at physical layer of the OSI model.
  - The **repeater** allows to transfer the data through large area distance
- **Hub**
  - It is a multiport device that allows multiple computers to communicate with each other over a network.
  - It is a non-intelligent network device that sends message to all ports( i.e. Broadcast)
  - **Types of Hubs**
    - **Active Hub** –
      - It strengthens the signal and may boost noise too.
      - It is relatively costlier than passive hub.
    - **Passive Hub** –
      - It repeat/copy signals.
      - It is relatively cheaper than active hub.
- **Switch**
  - Network Switch or switch is also a network multiport device that allow multiple computers to connect together.
  - Network switch inspects the packet, determine source and destination address and route the packet accordingly.
  - It operates at Data Link Layer (layer 2) of OSI model.
- **Bridge**
  - It connects multiple network segments having same protocol
  - It works at Data Link Layer (Layer 2).
  - Bridge does not simply broadcast traffic from one network.
  - Bridges use bridge table to send frames across network segments.
  - It also improves the overall network performance.
- **Router**
  - A router is a device that connects two or more packet-switched networks or sub networks.
  - It serves two primary functions:
    - Managing traffic between these networks by forwarding data packets to their intended IP addresses, and
    - Allowing multiple devices to use the same Internet connection.
  - It connects LANs (local area networks) and WANs (wide area networks).
  - It operates on layer 3 or 4 in OSI model
- **Gateway**
  - ❖ It is simply a device or hardware that acts as a "gate" between the networks.
  - ❖ It connects two networks with different transmission protocols together.
  - ❖ It converts information, data or other communications from one protocol or format to another.
  - ❖ It operates on layer 5 of OSI model
- **RJ45**
  - ❖ It stands for Registered Jack.
  - ❖ It is common interface to connect Twisted Pair Cable.
  - ❖ It is used for Ethernet and Token Ring Network.
- **Ethernet Card**



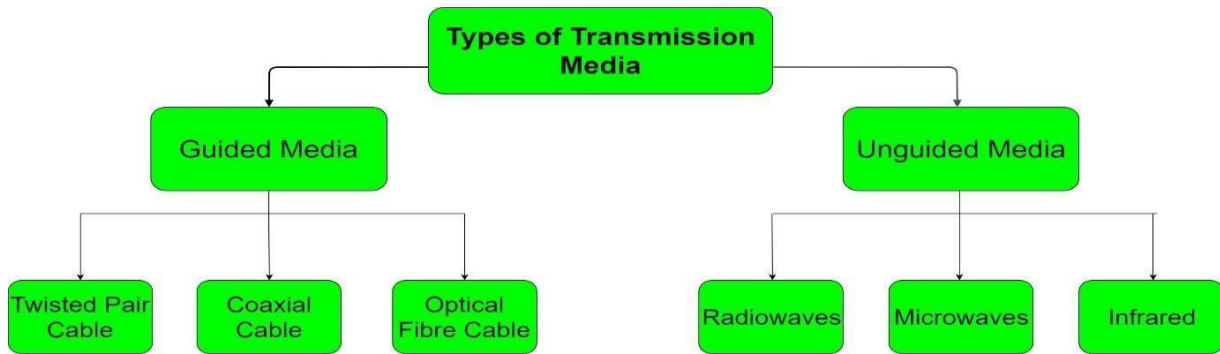
- ❖ It also known as NIC card.
- ❖ It enables a computer to access an Ethernet network (LAN)
- ❖ It has MAC id which gives it unique identity in the network.
- **Wi-Fi card**
  - ❖ It is also known wireless network adaptor.
  - ❖ It is a wireless network technology that allows devices to communicate over wireless signals.
  - ❖ It uses radio waves for the communication
- **Difference between Router and Switch**
  - ❖ A network switch forwards data packets between groups of devices in the same network, whereas a router forwards data between different networks.
- **Difference between a Router and a Modem**
  - ❖ A router forms networks and manages the flow of data within and between those networks, while a modem connects those networks to the Internet.
- **Difference between a Router and Gateway**
  - ❖ A gateway is a concept while a router is a device that implements a gateway.

Router	Gateway
It ensure that data packets are switched to the right address with the best route.	To connect two networks of different protocols as a translator
It routes the data packets via similar networks	It connects two dissimilar networks
It supports dynamic Routing.	It does support dynamic Routing.

### **Type of Network**

- **PAN**
  - It stands for Personal Area Network.
  - It is a computer network formed around a person.
  - It generally consists of a computer, mobile, or personal digital assistant.
  - Appliances use for PAN: cordless mice, keyboards, and Bluetooth systems.
  - PAN includes mobile devices, tablet, and laptop.
- **LAN**
  - ❖ It is a group of computer and peripheral devices which are connected in a limited area such as room, building & campus.
  - ❖ Higher Data Speed.
  - ❖ LANs are in a narrower geographic scope (up to 1 Km).
  - ❖ It is a private network.
- **MAN**
  - ❖ A **Metropolitan Area Network** or MAN is consisting of a computer network that span across a city.
  - ❖ It mostly covers towns and cities in a maximum 50 km range.
  - ❖ The dual bus in MAN network provides support to transmit data in both directions concurrently.
- **WAN**
  - ❖ It connects device across globe.
  - ❖ It uses public network
  - ❖ Internet
  - ❖ BSNL

## Network Media



### ➤ **Guided or Wired**

- ❖ Telephone (T1) cable
- ❖ Twisted pair cable
  - STP (Shielded Twisted Pair)
  - UTP (Unshielded Twisted Pair)
- ❖ Co-axial cable
- ❖ Optical Fiber/Fibre

### ➤ **Unguided or Wireless**

- ❖ Infrared
- ❖ Radio Wave
- ❖ Microwave
- ❖ Bluetooth
- ❖ Satellite

### **Twisted Pair Cable**

- A twisted pair cable comprises of two separate insulated copper wires, which are twisted together and run in parallel.
  - ❖ A STP (Shielded Twisted Pair) cable has a fine wire mesh surrounding the wires to protect the transmission
  - ❖ UTP (Unshielded Twisted Pair) cable does not have a fine wire mesh.
- It is also known as Cat# cable where # denote number. e.g., Cat6
- Connector: RJ 45

### **Co-axial Cable**

- Coaxial cabling has a single copper conductor at its center, and a plastic layer that provides insulation between the center conductor and a braided metal shield.
- Connector: BNC (Bayonet Neill-Concelman)

### **Optical Fibre**

- An optical fiber is a flexible, transparent fiber made by drawing glass or plastic to a diameter slightly thicker than that of a human hair.
- It uses light for data transmission using total internal reflection.

### **Unguided Media or Wireless Media**

- No Physical media is used
- Less Secure
- Relatively low speed
- Can be used for longer distance
- Best suited for difficult terrain
- There is no need to acquire land rights

### **Radio Wave**

- Frequency – 3KHz – 1GHz
- Omni-Directional

- Penetrate obstacle
- Antenna of sender and receiver should not be aligned

### **Infrared**

- 300GHz to 400THz
- Line of sight- antenna of sender and receiver must be aligned
- Short distance communication
- It cannot penetrate obstacle – best suited for indoor
- Secure
- Support high data rate
- TV Remote

### **Microwave**

- 1GHz to 300 GHz
- Line of sight- antenna of sender and receiver must be aligned
- Cannot penetrate obstacles
- Rain or other disturbance cause issue with Microwave
- Types of microwave propagation
  - ❖ Terrestrial Microwave propagation
  - ❖ Satellite Microwave propagation

### **Bluetooth**

- It also uses radio waves
- 2.4 GHz
- Range 10mtr
- Short distance

## **Topology**

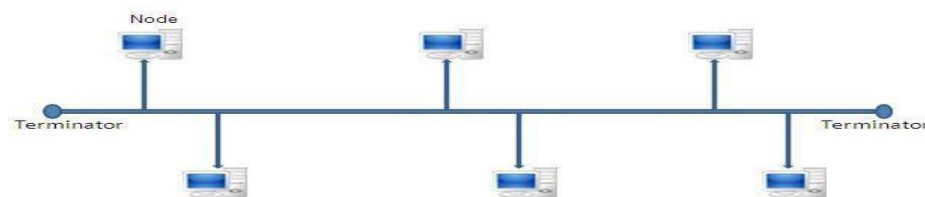
- Physical and Logical arrangement of nodes in the network is called Network Topology.

### **Types of Topologies**

- Bus
- Ring
- Star
- Tree
- Mess
- Hybrid

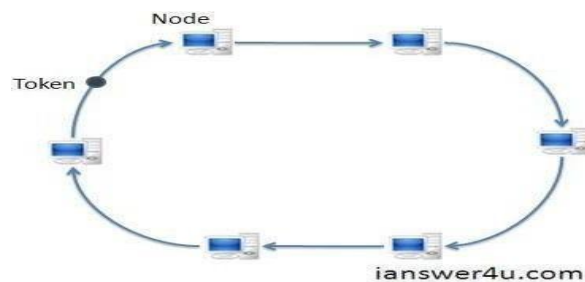
### **Bus Topology**

- In Bus Topology all the nodes are connected to single cable or backbone
- Both the end has terminators.



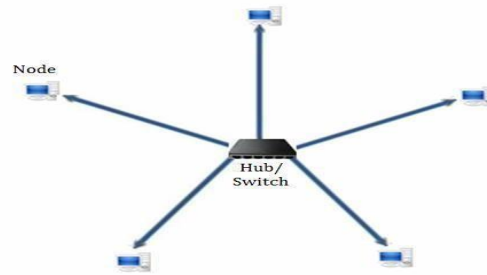
### **Ring Topology**

- In Ring Topology all the nodes are connected to each-other to form a loop.
- Each workstation is connected to two other components on either side
- It communicates with these two adjacent neighbors.
- Data is sent and received using Token.



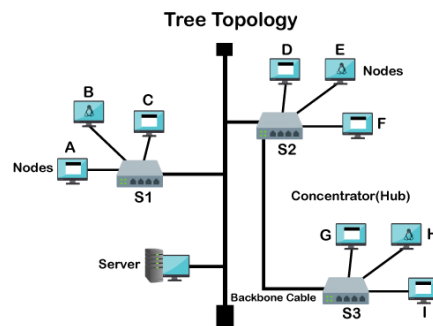
## Star Topology

- In Star Topology all the nodes are connected to a central device called Hub/Switch.
- All communication is controlled by the central Device (Hub/Switch)



## Tree Topology

- In Tree Topology, the devices are arranged in a tree fashion similar to the branches of a tree.
- It has multilayer architecture.



## Protocol

- It is set of rules or standard that governs communication.

**Types of Protocol** (Broadly can be kept in two suites of Protocols vis. **TCP/IP** or **OSI**)

- FTP
- HTTP/HTTPS
- IMAP
- POP3
- SMTP
- PPP
- TELNET
- VoIP

## **TCP/IP – Transmission Control Protocol/ Internet Protocol**

- It is a protocol suite consist of two protocols **Transmission Control Protocol and Internet Protocol**.
- TCP ensures reliable transmission or delivery of packets on the network.

## **HTTP (Hyper Text Transfer Protocol)**

- It is is an application-layer protocol for transmitting hypermedia documents, such as HTML.

## **HTTPS (Secure Hyper Text Transfer Protocol)**

- It is an extension of HTTP protocol for transmitting hypermedia documents, such as HTML securely over a network.
- It encrypts data to be sent using TLS (Transport Layer Security)/SSL (Secure Sockets Layer).

## **FTP (File Transmission Protocol)**

- It is used for the transfer of computer files among hosts over TCP/IP (internet).

## **Telnet (TERminal NETWork)**

- It is an application **protocol** that allows a user to communicate with a remote device.

## **SMTP (Simple Main Transfer Protocol)**

- It is used to send mail from mail client to mail server over internet.

### **POP3 (Post Office Protocol)**

- It provides mechanism for retrieving emails from a remote server for a mail recipient.
- POP3 downloads the email from a server to a single computer, then deletes the email from the server.

### **IMAP (Internet Message Access Protocol)**

- It is also used to retrieve mail from mail server to client over internet (TCP/IP).
- It allows access to mail from different device.

### **VoIP (Voice over IP)**

- It is also known as Internet Telephony or Internet calling.
- It allows to make voice calls using a broadband Internet connection instead of a regular (or analog) phone line.

## **Introduction to web services**

### **WWW:**

- The World Wide Web, commonly known as the Web, is an information system where documents and other web resources are identified by Uniform Resource Locators, which may be interlinked by hyperlinks, and are accessible over the Internet.
- The Web is not the same as the Internet: The Web is one of many applications built on top of the Internet.
- Tim Berners-Lee proposed the architecture World Wide Web in 1989.

### **Application of Internet Web 2.0:**

- The term web 2.0 is used to refer to a new generation of websites that are supposed to let people to publish and share information online.
- It aims to encourage the sharing of information and views, creativity that can be consume by the other users e.g.: YouTube

### **Web 3.0**

- It refers to the 3rd Generation of web where user will interact by using artificial intelligence and with 3-D portals.
- Web 3.0 supports semantic web which improves web technologies to create, connect and share content through the intelligent search and the analysis based on the meaning of the words, instead of on the keywords and numbers.

### **Hyper Text Markup Language (HTML):**

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- Various Tags are
  - **<html>** represents the root of an HTML document
  - **<head>** element is a container for metadata (data about data) and is placed between the **<html>** tag and the **<body>** tag.
  - **<title>** tag defines the title of the document
  - **<body>** defines the document's body.
  - **</br>** Line Break Tags
  - **<h1> <h2> .....</h6>** - Heading Tags - tags are used to define HTML headings.
  - **<font>** - defines font face, font size, and color of text etc.

### **Extensible Markup Language (XML)**

- XML stands for eXtensible Markup Language.
- XML was designed to store and transport data.
- XML was designed to be both human- and machine-readable.
- XML is a markup language much like HTML
- XML was designed to be self-descriptive

## The Difference between XML and HTML

HTML	XML
It designed to display the data	It is designed to carry data
Its tags are predefined	Its tags user defined
It is not case sensitive	It is case sensitive
It is static	It is dynamic
It is Markup Language	It is framework to define Markup language
Closing tags are not necessary in HTML	Closing tags are necessary in XML

### Domain names

- A domain name is a website's address on the Internet.
- Domain names are used in URLs to identify to which server belong a specific webpage.
- The domain name consists of a hierarchical sequence of names (labels) separated by periods (dots) and ending with an extension.

### URL

- Uniform Resource Locator (URL) is a text string that specifies where a resource (such as a web page, image, or video) can be found on the Internet.

### Website

- Website is a group of web pages, containing text, images and all types of multi-media files.

### Web browser-

- A web browser, or simply "browser," is an application used to access and view websites. Common web browsers include Microsoft Internet Explorer, Google Chrome, Mozilla Firefox, and Apple Safari.

### Web servers:

- A web server is a computer that stores web server software and a website's component files (e.g. HTML documents, images, CSS style sheets, and JavaScript files).

### Web hosting:

- Web hosting is an online service that enables you to publish your website or web application on the internet. When you sign up for a hosting service, you basically rent some space on a server on which you can store all the files and data necessary for your website to work properly.
- **Types of Web Hosting**
  - Shared Hosting - Web server shared among customers
  - Reseller Hosting – Become yourself as web host to sell others
  - Virtual Private Server – One server as multiple server for multiple websites
  - Dedicated Server- Entire web server is dedicated to same website

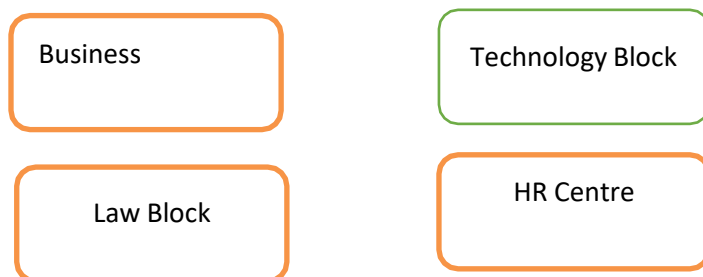
### Abbreviations:

Abbreviation	Full Form
LAN	Local Area Network
WAN	Wide Area Network
MAN	Metropolitan Area Network
FTP	File Transfer Protocol
SMTP	Simple Mail Transfer Protocol
IMAP	Internet Mail Access Protocol
MODEM	Modulator - Demodulator
WWW	World Wide Web
RPC	Remote Procedure Call
NFS	Network File System

HTML	Hyper Text Markup Language
DHTML	Dynamic Hyper Text Markup Language
HTTP	Hypertext Transfer Protocol
TCP/ IP	Transmission Control Protocol/ Internet Protocol
SLIP	Serial Line Internet Protocol
PPP	Point To Point Protocol
SIM	Subscriber's Identification Module
3G	3 <sup>rd</sup> Generation of Mobile Communication Technology
SMS	Short Message Service
EDGE	Enhanced Data rates for Global Evolution
E-mail	Electronic Mail
NFS	Network File System
WLL	Wireless in Local Loop
CDMA	Code Division Multiple Access
FRA	Fixed Radio Access
GSM	Global system For Mobile Communication
ARPANET	Advanced Research Project Agency Network
XML	Extensible Markup Language
HTML	Hypertext Markup Language
URL	Uniform Resource Locater
ISP	Internet Service Provider
DNS	Domain Name System
VSNL	Videsh Sanchar Nigam Limited
MTNL	Mahanagar Nigam Telephone Limited
WAIS	Wide Area Information Services
SLIP	Serial Line Internet Protocol
TCP	Transmission Control Protocol
TDM	Time Division Multiplexing
WDM	Wavelength Division Multiplexing
FDM	Frequency Division Multiplexing

Long Answer Question:

Quick Learn University is setting up its academic blocks at Prayag Nagar and planning to set up a network. The university has 3 academic blocks and one human resource Centre as shown in the diagram given below:



Centre-to-Centre distance between various blocks is as follows:

Law block to business block	40 m
Law block to technology block	80 m
Law block to HR block	105 m
Business block to technology block	30 m
Business block to HR block	35 m
Technology block to HR block	15 m

Number of computers in each of the buildings is as follows:

Law block	15
Technology block	40
HR Centre	115
Business block	25

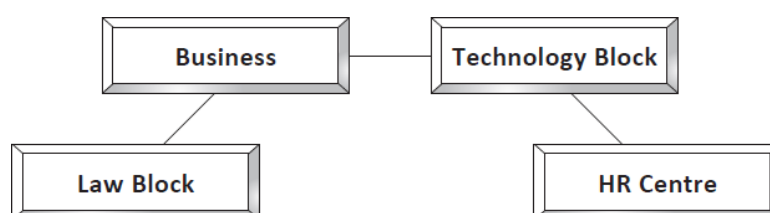
- Suggest a cable layout of connection between the blocks.
- Suggest the most suitable place to house the server of the organization with suitable reason.
- Which device should be placed/installed in each of these blocks to efficiently connect all the computers within these blocks?
- The university is planning to link its sales counters situated in various parts of the other cities. Which type of network out of LAN, MAN or WAN will be formed?
- Which network topology may be preferred in each of these blocks?

Most Important Notes Applicable to All Questions based on the above patterns, which may be kept in mind –

- In a Wing / Block / Lab connect all computers using a HUB or SWITCH. This is known as LAN connectivity. But now a day we use SWITCH only as it transmits data to the required destination only and thus faster in transmission.
- Computers in a Block are connected to the SWITCH/HUB using Cat 5 Cable. In place of Cat 5 Cable we can also use Fibre Optics cable if we require more speed and can afford cost.
- Server is placed where there is maximum number of computers as maximum traffic will be in this block only.
- An additional switch may also be used to connect the block's switches for faster transmission of data and this is also placed in the block where we have put the server. These switches can be connected to the additional switch using fibre optics since it will increase speed but cost has to be born.
- If the company desires to have connectivity to the Internet, Internet connection is taken and connected to switch. From here all other buildings and further computers can access the Internet connection. If the Company wants to connect these blocks to their office in some other distant place then we can connect these blocks with that office using an Internet Connection, Radio Wave Connection (if office is in hilly area), WAN connection if company can afford cost and requires security of data also. When WAN connectivity is established, switch is further connected to a ROUTER. Router is costly and filters the network traffic. It routes the data to the destinations using most optimal path.
- Connectivity between the block's switches is done using Cat 5 cables. If there is a distance of more than 100 m between any two blocks then a suitable REPEATER has to be placed to join Cat 5 cables as in a LAN we can use Cat 5 cable up to a maximum length of 100 m as the signal power attenuates with more distances. So REPEATER is used to regenerate the signal.

**So keeping in mind the above presumptions we answer**

(a)





- (b) HR centre because it consists of the maximum number of computers to house the server.
- (c) Switch should be placed in each of these blocks.
- (d) MAN
- (e) star

### Question and Answer

#### Objective Type Question (1 Mark Each)

1. What is the Use of Bridge?
  - (a) To Connect two LAN
  - (b) To Connect two LAN Segment
  - (a) To Connect Internet
  - (d) Amplify Signal
2. With the use of computer network, we can share
  - (a) Data
  - (b) Resources
  - (c) Both of the above
  - (d) None of the above
3. Which of the following is used in computer network
  - (a) Router
  - (b) Switch
  - (c) Bridge
  - (d) All of the Above
4. When we connect our mobile with Laptop then we call it\_\_\_\_\_
  - (a) MAN
  - (b) WAN
  - (c) PAN
  - (d) LAN
5. Mbps stands for\_\_\_\_\_
  - (a) Mega Byte Per Second
  - (b) Million Byte Per Second
  - (c) Megabits Per Second
  - (d) Maximum Bit Per Second
6. Full form of NIC
  - (a) New Interface Card
  - (b) Network Interface Card
  - (c) New Internet Card
  - (d) Network Internet card
7. Which device Convert Analog to Digital and vice-versa
  - (a) Hub
  - (b) Bridge
  - (c) Modem
  - (d) Switch
8. Arun wants to create his own website, which language he can use to create new website.
  - (a) XML
  - (b) HTML
  - (c) Python
  - (d) C++
9. Amit's computer is running slow due to virus. Then what he has to do.
  - (a) Format Computer
  - (b) Install Firewall
  - (c) Install Antivirus
  - (d) Do Nothing
10. Hari is copying Mohan's software without taking permission with Mohan. So which type of crime is Hari doing
  - (a) Cyber Bulling
  - (b) Cyber Stalking
  - (c) Cyber Theft
  - (d) Copyright Infringement

#### Answers:

- |                               |                            |
|-------------------------------|----------------------------|
| 1. To Connect two LAN Segment | 6. Network Interface Card  |
| 2. Both of the above          | 7. Modem                   |
| 3. All of the Above           | 8. HTML                    |
| 4. PAN                        | 9. Install Antivirus       |
| 5. Megabits Per Second        | 10. Copyright Infringement |

#### Very Short Answer Type Question (2 Marks each)

1. What is the difference between Packet switching and circuit switching techniques?

Circuit Switching	Packet Switching
Circuit switching requires a dedicated path before sending data from source to destination.	Packet switching does not require any dedicated path to send data from source to destination.

Circuit Switching	Packet Switching
It reserves the entire bandwidth in advance.	It does not reserve bandwidth in advance
No store and forward transmission	It supports store and forward transmission
Each packet follows the same route	A packet can follow any route

2. Write the full form of the following
  - a. FTP: File Transfer Protocol
  - b. TCP: Transmission Control Protocol
  - c. VoIP: Voice over Internet Protocol
  - d. SMTP: Simple Mail Transfer Protocol
3. Difference between LAN and WAN

LAN is private.	WAN can be private or public.
The full form of LAN is Local Area Network.	The full form of WAN is Wide Area Network.
The speed of LAN is higher.	The speed of WAN is slower.
The configuration and maintenance is easy in LAN.	The configuration and maintenance is harder than LAN.
LAN covers smaller areas like school, hospital, etc.	WAN covers a large area like a country.
Coaxial or UTP cable is the transmission medium used by LAN.	PSTN or satellite link is a transmission medium used by WAN.

4. Explain Any two Wired Media
  - a. **Twisted Pair Cable**  
A twisted pair cable comprises of two separate insulated copper wires, which are twisted together and run in parallel.
  - b. **Fibre Optics**  
Fibre optic cables are mainly used to transmit information over long distances with minimum loss. The information through optical fibres is transmitted in the form of light pulses. The core of the optical fibres is made of glass or plastic
5. Define Any Two Topologies
  - a. **Star**  
Star topology is a network topology in which each network component is physically connected to a central node such as a router, hub or switch. In a star topology, the central hub acts like a server and the connecting nodes act like clients
  - b. **Bus**  
A bus topology is a type of local area network configuration in which computers or terminals (also known as nodes) are connected to a single cable (as known as the backbone)
6. Write short note on the following:
  - (a) Bandwidth
  - (b) DNS

Answer: -

(a) **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. Normally, bandwidth is the difference of maximum and minimum frequency contained in the composite signals. Bandwidth is measured in Hertz (Hz).

(b) **DNS:** - A way to translate a URL (domain name) into IP address and vice-versa.

7. Write the name of the most suitable wireless communication channels for each of the following situations.
- Communication between two offices in two different countries.
  - To transfer the data from one mobile phone to another.

**Answer:**

- Satellite Communication
- Bluetooth or infrared whichever is supported by the phone.

### Short Answer Type Question (3 Marks each)

1. What is Cloud Computing? Name three types also

Cloud computing is the on-demand availability of computer system resources, especially data storage and computing power, without direct active management by the user. Large clouds often have functions distributed over multiple locations, each of which is a data centre.

Types of Cloud

- Private clouds**
- Public clouds**
- Hybrid clouds**

2. What is ARPANET? What is NSFNET?

ARPANET (Advanced Research Project Agency Network) is a project sponsored by US Department of Defense.

NSFNET, developed by the National Science Foundation, was a high-capacity network and strictly used for academic and engineering research.

### Long Answer Type Question (4 Marks each)

1. Define web browser and web server.

**Web Browser:** - A web browser is a software which is used for displaying the content on web page(s). It is used by the client to view websites. Examples of web browser: - Google Chrome, Firefox, Internet Explorer, Safari, Opera, etc.

**Web Server:** - A web server is a software which entertains the request(s) made by a web browser. A web server has different ports to handle different requests from web browser, like generally FTP request is handled at Port 110 and HTTP request is handled at Port 80. Example of web server is Apache.

2. Difference between HTML and XML

#### **HTML**

HTML stands for Hyper Text Markup Language.

HTML is static in nature.

HTML is a markup language.

HTML can ignore small errors.

HTML is not Case sensitive.

HTML tags are predefined tags.

#### **XML**

XML stands for extensible Markup Language.

XML is dynamic in nature.

XML provides framework to define markup languages.

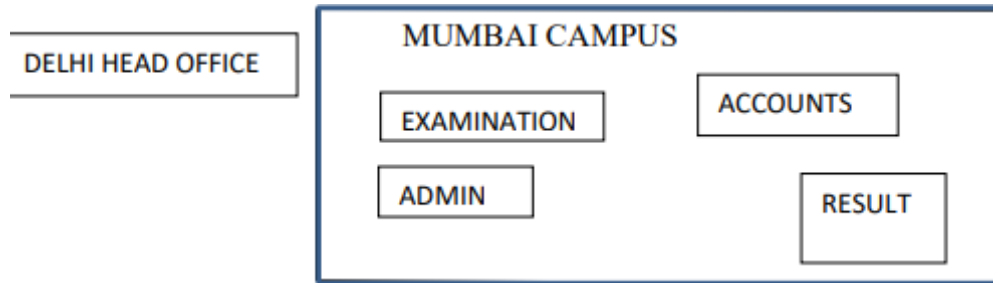
XML does not allow errors.

XML is Case sensitive.

XML tags are user defined tags.

**Very Long Answer Type Question (5 Marks each)**

- You as a network expert have to suggest the best network related solutions for their problems raised in (i) to (v), keeping in mind the distances between the buildings and other given parameters.



Shortest distances between various buildings:

ADMIN TO ACCOUNTS	55M
ADMIN TO RESULT	50M
ACCOUNTS TO RESULT	50M
ADMIN TO EXAMINATION	90M
DELHI Head Office to MUMBAI	2150M
ACCOUNTS TO EXAMINATION	55M
EXAMINATION TO RESULT	45M

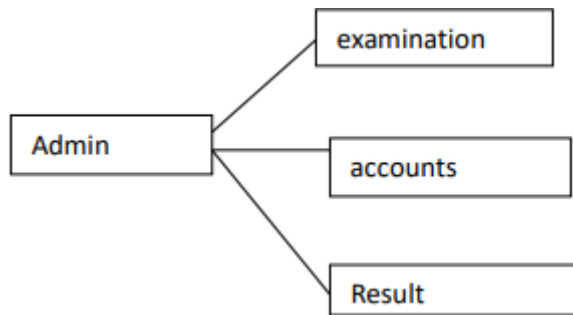
Numbers of computers installed at various buildings are as follows:

ADMIN	110
ACCOUNTS	75
EXAMINATION	40
RESULT	12
DELHI HEAD OFFICE	20

- Suggest the most appropriate location of the server inside the MUMBAI campus (out of the four buildings) to get the best connectivity for maximum number of computers. Justify your answer.
- Suggest and draw cable layout to efficiently connect various buildings within the MUMBAI campus for a wired connectivity.
- Which networking device will you suggest to be procured by the company to interconnect all the computers of various buildings of MUMBAI campus?
- Company is planning to get its website designed which will allow students to see their results after registering themselves on its server. Out of the static or dynamic, which type of website will you suggest?
- Which of the following will you suggest to establish the online face to face communication between the people in the ADMIN office of Mumbai campus and Delhi head office?
  - Cable TV
  - Email
  - Video conferencing
  - Text chat

**Ans:**

i. Server should be installed in Admin department as it has maximum number of computers.



ii.

Star topology 5 Admin examination accounts Result

iii. Hub/Switch

iv. Dynamic

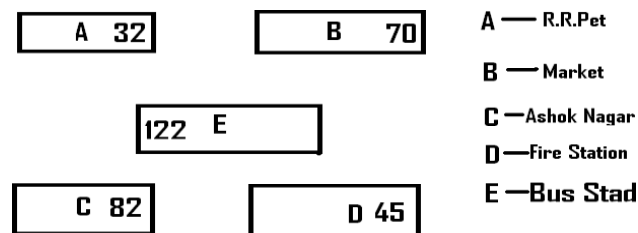
v. Video conferencing

**Additional Exercise:**

1. Which protocol is used to exchange files on Internet
2. What is the purpose of using a MODEM?
3. What type of address is the following?  
20:B9:F1:63:2F:FB
4. Identify the topologies from the followings:
  - (i) In it the nodes form a circular path for data to travel and each node is connected to two other nodes.
  - (ii) In it devices are connected through hub/switch, hub/switch is responsible for receiving and transmitting data from each node to destination.
5. A School would like to go in for network of all the computers. Which topology would you recommend and why?
6. What is communication channel? What choices do you have while choosing a communication channel for a network?
7. What do you mean by network topology? Name most popular topologies.
8. Akhil is transferring songs from his mobile to his friend's mobile via Bluetooth connection. Name the network used by Akhil.

**Case Based**

9. In Hyderabad, 5 ABC Bank branches are available. One is at RR Pet, other at Market, other at Ashok Nagar, other at Fire Station and the last one at Bus Stand. Higher official wants to keep a network between these 5 branches. The branch names (A to E) and the number of computers in each branch (given inside the rectangle) is given below.

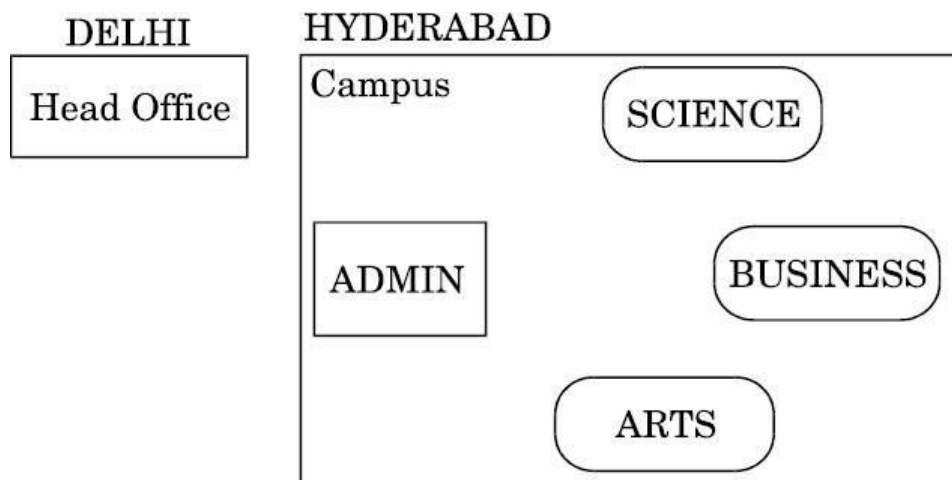


Distance between various buildings	
A to B	50 Mts
B to C	30 Mts
C to D	30 Mts
D to E	35 Mts
E to C	40 Mts
D to A	120 Mts
D to B	45 Mts
E to B	65 Mts

- (i) Suggest a possible cable layout for connecting the buildings.
- (ii) Suggest the most suitable place to install the server of this organization with a suitable reason
- (iii) Suggest the placement of the following devices with justification.
  - (a) Hub/Switch
  - (b) Modem
- (iv) The Bank wants to link its head Office in 'A' building to its main office at Mumbai.
  - (a) Which type of transmission medium is appropriate for such a link?
  - (b) What type of network this connection result into?

11. Xcelencia Edu Services Ltd. is an educational organization. It is planning to set up its India campus at Hyderabad with its head office at Delhi. The Hyderabad campus has 4 main buildings - ADMIN, SCIENCE, BUSINESS and ARTS. You as a network expert have to suggest the best network related solutions for their problems raised in (i) to (iv), keeping in mind the distances between the buildings and other given parameters.

[CBSE 2015 Main]



Shortest distances between various buildings:

ADMIN to SCIENCE	65 m
ADMIN to BUSINESS	100 m
ADMIN to ARTS	60 m
SCIENCE to BUSINESS	75 m
SCIENCE to ARTS	60 m
BUSINESS to ARTS	50 m
DELHI Head Office to HYDERABAD Campus	1600 Km

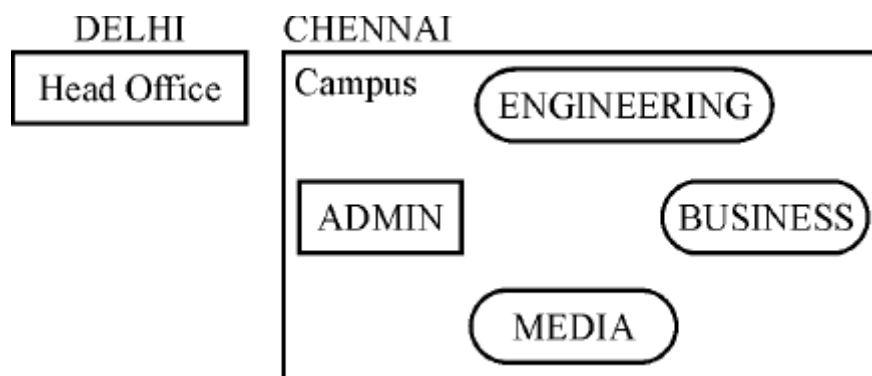
Number of computers installed at various buildings are as follows:

ADMIN	100
SCIENCE	85
BUSINESS	40
ARTS	12
DELHI Head Office	20

- (i) Suggest the most appropriate location of the server inside the HYDERABAD campus (out of the 4 buildings), to get the best connectivity for maximum number of computers. Justify your answer.
- (ii) Suggest and draw the cable layout to efficiently connect various buildings within the HYDERABAD campus for connecting the computers.
- (iii) Which hardware device will you suggest to be procured by the company to be installed to protect and control the internet uses within the campus?
- (iv) Which of the following will you suggest to establish the online face-to-face communication between the people in the Admin Office of HYDERABAD campus and DELHI Head Office?
  - a. Email
  - b. Text Chat
  - c. Video Conferencing
  - d. Cable TV

1. Perfect Edu Services Ltd. is an educational organization. It is planning to setup its India campus at Chennai with its head office at Delhi. The Chennai campus has 4 main buildings – ADMIN, ENGINEERING, BUSINESS and MEDIA. You as a network expert have to suggest the best network related solutions for their problems raised in (i) to (iv), keeping in mind the distances between the buildings and other given parameters.

[CBSE COMP 2015]



Shortest distances between various buildings:

ADMIN to ENGINEERING	55 m
ADMIN to BUSINESS	90 m
ADMIN to MEDIA	50 m
ENGINEERING to BUSINESS	55 m
ENGINEERING to MEDIA	50 m
BUSINESS to MEDIA	45 m

DELHI Head Office to CHENNAI Campus	2175 km
-------------------------------------	---------

Number of Computers installed at various buildings are as follows :

ADMIN 110	110
ENGINEERING 75	75
BUSINESS 40	40
MEDIA 12	12
DELHI Head Office 20	20

- (i) Suggest the most appropriate location of the server inside the CHENNAI campus (out of the 4 buildings), to get the best connectivity for maximum no. of computers. Justify your answer.
  - (ii) Suggest and draw the cable layout to efficiently connect various buildings within the CHENNAI campus for connecting the computers.
  - (iii) Which hardware device will you suggest to be procured by the company to be installed to protect and control the internet uses within the campus?
  - (iv) Which of the following will you suggest to establish the online face-to-face communication between the people in the Admin Office of CHENNAI campus and DELHI Head Office?
    - a. Email
    - b. Text Chat
    - c. Video Conferencing
    - d. Cable TV
10. Expand the following:
    - a) ARPANET
    - b) MAC
    - c) ISP
    - d) URI
  11. What do you understand by the term network?
  12. Mention any two main advantages of using a network of computing devices.
  13. Differentiate between LAN and WAN.
  14. Write down the names of few commonly used networking devices.
  15. Two universities in different States want to transfer information. Which type of network they need to use for this?
  16. Define the term topology. What are the popular network topologies?
  17. How is tree topology different from bus topology?
  18. Identify the type of topology from the following:
    - a) Each node is connected with the help of a single cable.
    - b) Each node is connected with central switching through independent cables.
  19. What do you mean by a modem? Why is it used?
  20. Explain the following devices:
    - a) Switch
    - b) Repeater
    - c) Router
    - d) Gateway
    - e) NIC
  21. Sahil, a class X student, has just started understanding the basics of Internet and web technologies. He is a bit confused in between the terms “World Wide Web” and “Internet”. Help him in understanding both the terms with the help of suitable examples of each.



## Database Management System

### **DATABASE:**

- May be defined as a collection of interrelated data stored together to serve multiple application
- It is computer based record keeping system.
- It not only allows to store but also allows us modification of data as per requirements

### **DBMS:**

- A DBMS refers to Database Management System
- It is software that is responsible for storing, manipulating, maintaining and utilizing database.
- A database along with a DBMS is referred to as a database system.
- There are various DBMS software available in the market. e.g. Oracle, MS SQL Server, MySQL, Sybase, PostgreSQL, SQLite

### **Purpose of DBMS:**

- Reduce Data redundancy
- Control Data Inconsistency
- Sharing of data
- Ensure data integrity
- Enforce standard

### **Relational Database Model:**

- In relational database model data is organized into table (i.e. rows and columns).
- These tables are also known as relations.
- A row in a table represents relationship among a set of values.
- A column represents the field/attributes related to relation under which information will be stored.
- For example if we want to store details of students then : Roll, Name, Class, Section, etc. will be the column/attributes and the collection of all the column information will become a Row/Record

### **Sample Tables:**

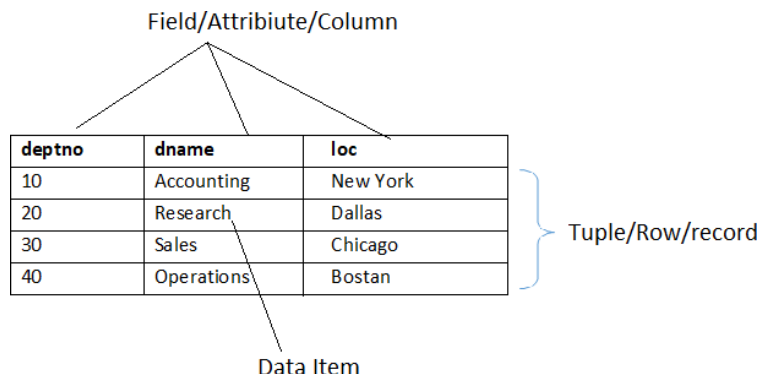
EMPLOYEE:

Dept:

Deptno	dname	Loc
10	Accounting	New York
20	Research	Dallas
30	Sales	Chicago
40	Operations	Bostan

### **Component of a table:**

- Data Item: smallest unit of named data. It represent one type of information and often referred to as a field or column information
- Record : collection of data items which represent a complete unit of information
- Table: collection of all Rows and Columns.



### Relational Data Model:-

Data is organized in two-dimensional tables called **relations**. The tables or relations are related to each other.

**Characteristics** of relational database are:-

- Data is arranged into rows and columns,
- Each relation is represented as a table.
- Every row in a table represents a single entity.
- At any given row or column position, there is one and only one value.

**Various Terms Used in Relational Model:** - A **relational database** is a type of database that stores and provides access to data points that are related to one another.

**Basic Terminologies related to a Relational Database:-**

- **Entity:** An entity is something that exists and an object which can be distinctly identified. *For example*, student entity, employee entity,
- **Attribute:** The term attribute is also used to represent a column.
- **Tuple:** Each row in a table is known as tuple.
- **Cardinality of Relation:** It is the number of records or tuples in a relation.
- **Degree of Relation:** Number of columns or attributes is known as degree of a relation.
- **Domain of Relation:** It defines the kind of data represented by the attribute.
- **Body of the Relation:** It consists of an unordered set of 0 or more tuples.

### Concept of Keys

- In relation each record must be unique i.e. no two identical records are allowed in the Database.
- A key attribute identifies the record and must have unique values. There are various types of Keys:

#### Primary key:

- A set of one or more attribute that can identify a record uniquely in the relation is called Primary Key.
- There can be only one primary key in a table
- Allows only distinct (no duplicate) values and also forces mandatory entry (NOT NULL) i.e. we can't leave it blank.

#### Candidate Key

- In a table there can be more than one attribute which uniquely identifies a tuples in a relation. These columns are known as candidate key as they are the candidate for primary key.
- Among these database analyst select only one attribute as a primary key.

### Alternate Key

- In case of multiple candidate keys, one of them will be selected as Primary Key and rest of the column will serve as Alternate Key.
- A Candidate Key which is not a primary key is an Alternate Key.

### Foreign key

- Used to create relationship between two tables.
- It is a non-key attribute whose value is derived from the Primary key of another table.
- Primary Key column table from where values will be derived is known as Primary Table or Master Table or Parent Table and Foreign key column table will be Foreign Table or Detail Table or Child table.

### Example:

#### EMPLOYEE

EMPNO	ENAME	GENDER	DEPTNO	SALARY	COMM
1	ANKITA	F	10	20000	1200
2	SUJEET	M	20	24000	
3	VIJAYA	F	10	28000	2000
4	NITIN	M	30	18000	3000
5	VIKRAM	M	30	22000	1700

Child Table

#### DEPARTMENT

DEPTNO	DNAME	LOCATION
10	HR	NEW YORK
20	ACCOUNTS	BRAZIL
30	SALES	CANADA
40	IT	INDIA

Parent Table

From the Above table definition we can observe that the DEPTNO column of EMPLOYEE table is deriving its value from DEPTNO of table DEPARTMENT. So we can say that the DEPTNO of EMPLOYEE table is a foreign key whose value is dependent upon the Primary key column DEPTNO of table DEPARTMENT.

### REFERENTIAL INTEGRITY:

- Used to ensure relationships between records in related tables are valid and user don't accidentally delete or change the related data.
- Referential integrity can be applied when:
  - The master table's column is a Primary Key or has a unique index
  - The related fields have the same data type
  - Both tables must belong to same database.
- When referential integrity is enforced using Foreign Key you must observe the following rules:
  - You cannot enter a value in Child Table which is not available in Master Table's Primary key column. However you can enter NULL values in foreign key
  - You cannot delete a record from Master Table if matching record exists in related table.
  - You cannot modify or change the Primary Key value in Master table if its matching record is present in related table.

### Structured Query Language

- It is a language that enables you to create and operate on relational databases
- It is the standard language used by almost all the database s/w vendors.
- Pronounced as SEQUEL
- It is portable i.e. it is compatible with most of the database.
- It is not a case sensitive language.
- It is very easy to learn.

## SQL – features

- Allows creating/modifying a database's structure
- Changing security settings for system
- Permitting users for working on databases or tables
- Querying database
- Inserting/modifying/deleting the database contents

## Classification of SQL

- DDL (Data Definition Language)
- DML (Data Manipulation Language)
- DCL (Data Control Language)
- TCL (Transaction Control Language)

## Data Definition Language:

- It allows us to create database objects like creating a table, view or any other database objects.
- The information about created objects are stored in special file called DATA DICTIONARY.
- DATA DICTIONARY contains metadata i.e. data about data.
- The commands of DDL are –
  - CREATE – To create a new database object.
  - ALTER – To modify existing database object.
  - DROP – To permanently remove existing database object.

## Data Manipulation Language:

- It allows to perform following operation on data in the table
  - Retrieval of information stored in table
  - Insertion of new data in table
  - Modification of existing data in table
  - Deletion of existing data from table
- The commands of DML are
  - SELECT – To retrieve data from the table
  - INSERT – To insert new tuple /row in the table
  - UPDATE – To update existing tuple/row from the table
  - DELETE – To delete existing tuple/row from the table

## Exercise:

1. What is Database? What are the advantages of Database System?
2. What is DDL and DML? Give examples of command belonging to each category
3. What is the difference between Primary key and Candidate key?
4. What is Primary Key? What are the restrictions imposed by Primary Key? How many primary key can be applied on a Table?
5. What is Degree and Cardinality of table?
6. Explain the purpose of DDL and DML commands used in SQL. Also give two examples of each.

## Introduction of MYSQL

### Brief history of MySQL:

- MySQL is freely available open source RDBMS
- It can be downloaded from [www.mysql.org](http://www.mysql.org)
- In MySQL information is stored in Tables.
- Provides features that support secure environment for storing, maintaining and accessing data.
- It is fast, reliable, scalable alternative to many of the commercial RDBMS today.

## **MYSQL DATABASE SYSTEM:**

- MySQL database system refers to the combination of a MySQL server instance and MySQL database.
- It operates using Client/Server architecture in which the server runs on the machine containing the database and client connects to server over a network
- MySQL is a multiuser database system, meaning several users can access the database simultaneously.

### **The Server**

- Listens for client requests coming in over the network and access the database as per the requirements and provide the requested information to the Client.

### **The Client**

- Are the programs that connect to MySQL server and sends requests to the server and receives the response of Server. Client may be the MySQL prompt or it may be Front-end programming which connects to server programmatically like connecting to MySQL using Python Language or Java or any other language.

## **FEATURES OF MYSQL:**

- Speed - MySQL runs very fast.
- Ease of Use - Can be managed from command line or GUI
- It is available free of cost. It is Open Source Software.
- Query language Support -Supports SQL
- Portability – Can be run on any platform and supported by various compilers.
- Data Types - supports various data types like Numbers, Char etc.
- Security -Offers privileges and password systems that is very flexible and secure.
- Scalability and Limits -Can handle large databases. Some of real life MySQL databases contain millions of records.
- Connectivity - Clients can connect to MySQL using drivers
- Localization -The server can provide error message to client in many language

## **SQL and MYSQL:**

- SQL stands for Structured Query Language.
- It is a language that enables you to create and operate on relational databases.
- MySQL uses SQL in order to access databases.
- It is the standard language used by almost all the database s/w vendors.

## **MYSQL Elements**

- Literals
- Data types
- Null
- Comments

### **Literals**

- It means the fixed value or constant value. It may be of character, numeric or date time type.
- Character and date/time literals are always in single quotation marks whereas numeric literals must be without single quotation marks.
- For example – ‘Virat’, 12, 12.56, ‘04-20-2018’.
- Date and time values are always in the format YYYY-MM-DD HH:MI:SS.
- Special character like quotes are always written by preceding it back-slash(\). For example if we want to store value as Tom’s Cat then it should be written as Tom\'s Cat

### **Data Type**

- Means the type of value and type of operation we can perform on data. For example, on numeric value we can store numbers and perform all arithmetic operations and so on.
- MySQL support three categories of data types:
- Numeric

- Date and time
- String types

➤ Numeric Data Types

Data type	Description
INT	Numbers without decimal. Store up to 11 digits. -2147483648 to 2147483647
FLOAT(M,D)	Real numbers i.e. number with decimal. M specify length of numeric value including decimal place D and decimal symbol. For example if it is given as FLOAT(8,2) then 5 integer value 1 decimal symbol and 2 digit after decimal TOTAL – 8. it can work on 24 digits after decimal.
DOUBLE(M,D)	Real numbers with more precision up to 53 place after decimal.
DECIMAL	It is used to store exact numeric value that preserves exact precision for e.g. money data in accounting system. <b>DECIMAL(P,D) means P no. of significant digits (1-65), D represent no. of digit after decimal(0-30), for e.g DECIMAL(6,2) means 4 digit before decimal and 2 digit after decimal. Max will be 9999.99</b>

**Date and Time Types**

- **DATE**-A date in YYYY-MM-DD format between 1000-01-01 to 9999-12-31.
- **DATETIME**-Combination of date and time. For example to store 4th December 2018 and time is afternoon 3:30 then it should be written as – 2018-12-04 15:30:00
- **TIMESTAMP** - Similar to DATETIME but it is written without hyphen for example the above date time is stored as 20181204153000
- **TIME**-To store time in the format HH:MM:SS
- **YEAR(M)** -To store only year part of data where M may be 2 or 4 i.e. year in 2 digit like 18 or 4 digit like 2018.

**String Types**

- **CHAR(M)**
  - Fixed length string between 1 and 255.
  - It always occupy M size, irrespective of actual number of characters entered.
  - Mostly use in the case where the data to be insert is of fixed size like Grade (A,B,C,..) or Employee code as E001, E002, etc
- **VARCHAR(M)**
  - Variable length string between 1 and 65535 (from MySQL 5.0.3) , earlier it was 255
  - It takes size as per the data entered for example with VARCHAR(20) if the data entered is MOBILE then it will take only 6 byte.
  - It is useful for the data like name, address where the number of character to be enter is not fixed.
- **Difference between CHAR & VARCHAR**

Char	varchar
Fixed length string	Variable length string
Fast, no memory allocation every time	Slow, as it take size according to data so every time memory allocation is done
It takes more memory	It takes less space

## Simple Queries in SQL

- **Show Databases**- This command is used to list all databases on MySQL Server
- **Use <dbname>**- This command is used to change/select/open given database  
e.g. To open a database named 'test'  
>>>**use test**
- **Show tables** – This command will list all the tables from current database. If no database is selected it will generate error.
- **Select database ()** – This command will display the name of current database.
- **Desc <tablename> or Describe <tablename>** - This command will be used to display the structure of the table.
- **Create database <dbname>** - This command is use to create a new database.  
For example – **create database mydb** will new database **mydb**  
Above command will generate error, if database already exist. To supress the error following command can be used  
**create database if not exist mydb**
- **Drop database <dbname>** - This command is used to remove existing database.  
For example – **drop database mydb** will permanently delete **mydb** database.  
Above command will generate error, if database already exist. To supress the error following command can be used  
**drop database if exist mydb**
- **Create table <tablename>** - This command / statement is used to create a new table in a database. The syntax the is as follows:  
CREATE TABLE table\_name (  
    column1 datatype[(size)] [constraint] ,  
    column2 datatype[(Size)] [constraint],  
    column3 datatypeI[(size)] [constraints] ,  
    ....  
);  
Here size and constraints are optional.  
e. g.  
**CREATE TABLE pet (**  
    **name VARCHAR(20),**  
    **owner VARCHAR(20),**  
    **species VARCHAR(20),**  
    **sex CHAR(1),**  
    **birth DATE, death DATE**  
);
- **Droptable <tablename>** - This command is used to permanently delete the table from database.  
For example, **drop table pet;** will delete **pet** table from database
- **Alter table <tablename>** - This command is used to modify the structure of existing table such as adding new column, removing existing column, rename or changing data type, size and constraints.
  - **Adding new column to exiting table**  
**Syntax :**  
Alter table <tablename>  
Add [column] column\_name datatype [(size)];
  - **Removing new existing column from the table**  
**Syntax:**  
Alter table <tablename>  
drop column <column\_name>;
  - **Changing datatype/size of the column**  
**Systax:**

- alter table <tablename>
  - modify column <colname> datatype(size);
  - **Renaming column name**
  - Alter table tablename
  - Change old\_columnname new\_column\_name datatype (size)
- **Update <tablename>** - This command is used to update data from the table
- Syntax:
- UPDATE table\_name SET column\_name=new\_value, column2\_name=new\_value
- WHERE condition;
- e.g. UPDATE emp set sal=sal+100 where ename ='Scot'
- The above query will update salary of Scot by 100.
- If where clause is not given, it will update/modify the value of column from each row.
- **Inserting into table\_name** – This command is used to add new row to the table
- Syntax :
- INSERT INTO table\_name VALUES (list of values)
- **Select data from Table using Select statement**
- Syntax:

Select clause	select */column_list
from clause	from table(s)
where clause	where condition
group by clause	group by column_name
having clause	having condition
order by clause	order by column_name asc/desc
limit clause;	limit m,n;

- **Selecting/displaying entire data from the table**
- Syntax:
- SELECT \* From tablename;
- or
- SELECT ALL column\_name FROM tablename;
- **Selecting/displaying data from specific column**
- Syntax:
- SELECT column1, column2,column3, .... FROM tablename;
- **Giving describing name for column using column alias**
- Syntax:
- SELECT \*/col\_name as 'alias1' FROM tablename;
- **Removing duplicate value from the column using distinct clause**
- Syntax:
- SELECT DISTINCT column\_name FROM tablename;
- Inserting text in the query
- Syntax:
- SELECT 'text' FROM tablename;
- Number of row in result depend on total number of rows in the table
- Performing simple calculation in the query
- Syntax:
- Select 2+4;
- All operation can be performd such as addition (+), subtraction (-), multiplication (\*), division (/) and remainder (%)
- Performing calculation on column
- Syntax:
- SELECT column1 operator value, column2.... FROM Tablename
- SELECT ename, sal, sal\*12 'Annual Salary' from emp;
- Working with null value



Any operation on NULL will result into NULL, MySQL provide ifnull() function to work with null value. If column contain null value it is replace value given value, otherwise display original value.

Syntax:

If null(column, value\_to\_replace')

```
mysql> select ename, sal, comm, ifnull(comm, 'N/A') from emp;
+-----+-----+-----+-----+
| ename | sal   | comm  | ifnull(comm, 'N/A') |
+-----+-----+-----+-----+
| SMITH | 800.00 | NULL  | N/A                  |
| ALLEN | 1600.00 | 300.00 | 300.00              |
| WARD  | 1250.00 | 500.00 | 500.00              |
| JONES | 2975.00 | NULL  | N/A                  |
| MARTIN | 1250.00 | 1400.00 | 1400.00             |
| BLAKE | 2850.00 | NULL  | N/A                  |
| CLARK | 2450.00 | NULL  | N/A                  |
| SCOTT | 3000.00 | NULL  | N/A                  |
| KING  | 5000.00 | NULL  | N/A                  |
| TURNER | 1500.00 | 0.00 | 0.00                 |
| ADAMS | 1100.00 | NULL  | N/A                  |
| JAMES | 950.00 | NULL  | N/A                  |
| FORD  | 3000.00 | NULL  | N/A                  |
| MILLER | 1300.00 | NULL  | N/A                  |
+-----+-----+-----+-----+
14 rows in set (0.00 sec)
```

Here, comm column contains null value which is replaced by 'N/A'.

➤ Restricting rows using where clause :

Where clause in the query will restrict number of rows in the output based on condition.

Syntax:

SELECT \*/column list

FROM Tablename

where condition;

**Condition (column\_name operator expression)**

**Relational Operator**

> greater than                      < less than                      >= greater than equal to  
 <= less than equal to              = equal                          != or <> not equal to

**Logical Operator**

And – evaluated true if all the logical expression is true otherwise false.

Or - evaluated true if any the logical expression is true otherwise false.

Logical operator is used to combine two or more logical expression,

**Membership Operator**

In, Not in

The IN operator allows you to specify multiple values in a WHERE clause.

The IN operator is a shorthand for multiple OR conditions.

**Comparing NULL**

is null, is not null

NULL (Absence of value) value cannot be compared using Relational operator. The above statement is used to check whether column contains NULL or not.

**Range Operator**

Between

**Pattern Matching**

Like

Not Like

Like clause is used to match pattern using two wild card characters

\_ (underscore) – single unknown character

% (modulo) - Zero or more unknown characters

e.g.

words starting with 't'

't%'

words ending with 't'

'%t'

words containing 't' -

'%t%'

word with 't' as second letter -

'\_t%'

words with 't' as third last character -

'%t\_\_'

words containing four letter and 't' as second letter -

'\_\_t\_\_'

➤ Ordering data using ORDER BY clause

- ORDER BY clause is used to arrange data in ascending/descending order
- Ordering can be done on more than one column, in this case data is arranged according first column and then second column and onwards.
- By default data is arranged in ascending order, in no ordering information (asc/desc) is given.

**Syntax:**

SELECT \*/col\_list FROM tablename

ORDER BY col1 asc/desc, col2 asc/desc;

Aggregate Functions

- An aggregate function performs a calculation on multiple values and returns a single value.
- These function work on multiple rows collectively return single value.
- List of Aggregate functions are

- max() : return maximum value in set of value

```
mysql> select max(sal) from emp;
+-----+
| max(sal) |
+-----+
| 5000.00 |
+-----+
1 row in set (0.01 sec)
```

- min() – return minimum value from the set of values

```
mysql> select min(sal) from emp;
+-----+
| min(sal) |
+-----+
| 800.00 |
+-----+
1 row in set (0.00 sec)
```

- avg() –return average value in set of non-null values

```
mysql> select avg(sal) from emp;
+-----+
| min(sal) |
+-----+
| 800.00 |
+-----+
1 row in set (0.00 sec)
```

- sum() - Return the summation of all non-NULL values of the set of values.

```
mysql> select sum(sal) from emp;
+-----+
| sum(sal) |
+-----+
| 29025.00 |
+-----+
1 row in set (0.01 sec)
```

- count() - Return the number of rows in a group

- Count(\*) – return number of rows, including rows with NULL

```
mysql> select count(*) from emp;
+-----+
| count(*) |
+-----+
| 14 |
+-----+
1 row in set (0.00 sec)
```

- Count(column\_name) - return number of rows, excluding rows with NULL for the given column

```
mysql> select count(comm) from emp;
+-----+
| count(comm) |
+-----+
|          4 |
+-----+
1 row in set (0.00 sec)
```

- Count (distinct column\_name) – return number of rows with duplicate removed

```
mysql> select count(distinct job) from emp;
+-----+
| count(distinct job) |
+-----+
|          5 |
+-----+
1 row in set (0.01 sec)
```

In above example there are 14 rows in the EMP table, but the distinct clause only considers unique values.

### ➤ Group By Clause

- It is used in a SELECT statement to collect data across multiple records and group the results by one or more columns.

Syntax:

```
SELECT column_name, aggregate_function
FROM table_name
GROUP BY column_name
```

```
mysql> select job, max(sal) from emp group by job;
+-----+-----+
| job      | max(sal) |
+-----+-----+
| ANALYST  | 3000.00  |
| CLERK    | 1300.00  |
| MANAGER  | 2975.00  |
| PRESIDENT| 5000.00  |
| SALESMAN | 1600.00  |
+-----+-----+
```

In above example salary is grouped on job and maximum salary from each job is displayed.

- Select clause involving group by clause can contain column present in group by clause, aggregate function or no column. Otherwise it will return random data from other columns as given below.

```
mysql> select ename, job, max(sal) from emp group by job;
+-----+-----+-----+
| ename | job      | max(sal) |
+-----+-----+-----+
| SCOTT | ANALYST  | 3000.00  |
| SMITH | CLERK    | 1300.00  |
| JONES | MANAGER  | 2975.00  |
| KING  | PRESIDENT| 5000.00  |
| ALLEN | SALESMAN | 1600.00  |
+-----+-----+-----+
```

### ➤ Having clause –

- Having clause is used to place condition on aggregate function in conjunction with group by clause.
- Having clause is placed after where clause in select statement.

**Syntax:**

```
SELECT column_name, aggregate_function(col_name)
FROM table
WHERE condition
```

GROUP BY column\_name

HAVING aggregate\_function(column\_name) operator expression;

```
mysql> select deptno, max(sal), count(*) from emp group by deptno;
+-----+-----+-----+
| deptno | max(sal) | count(*) |
+-----+-----+-----+
|      10 | 5000.00 |         3 |
|      20 | 3000.00 |         5 |
|      30 | 2850.00 |         6 |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

The above query will display deptno, maximum salary and number of employees from each department.

The query given below display deptno, maximum salary and number of employees from those department which have maximum salary greater than equal to 3000.

```
mysql> select deptno, max(sal), count(*) from emp
-> group by deptno having max(sal)>=3000;
+-----+-----+-----+
| deptno | max(sal) | count(*) |
+-----+-----+-----+
|      10 | 5000.00 |         3 |
|      20 | 3000.00 |         5 |
+-----+-----+-----+
```

As condition is on aggregate function max(), where clause can't be used in this case.

Exercise Questions:

1. What is MySQL used for? Abhay wants to start learning MySQL. From where can he obtain the MySQL software ?
2. In the table "Student", Priya wanted to increase the Marks(Column Name: Marks) of those students by 5 who have got Marks below 33. She has entered the following statement:
3. SELECT Marks+5 FROM Student WHERE Marks<33;  
Identify errors(if any) in the above statement. Rewrite the correct SQL statement.
4. Write SQL statement to add a column "COUNTRY" with data type and size as VARCHAR(70) to the existing table named "PLAYER". Is it a DDL or DML or TCL command ?
5. Table Student has the columns RNO and SCORE. It has 3 rows in it. Following two SQL statements were entered that produced the output (AVG(SCORE) as 45 and COUNT(SCORE) as 2) :

- (i) AVG(SCORE)
- (ii) Count(score)

6. 'Employee' table has a column named 'CITY' that stores city in which each employee resides. Write SQL query to display details of all rows except those rows that have CITY as 'DELHI' or 'MUMBAI' or 'CHANDIGARH'.
7. How is a database related to a table ?
8. Mr. Sen has to create a table named 'Employee' with Columns to store EmpID, Name, Designation, Age and Salary. EmpID is the Primary key and Name cannot be NULL. Some of the rows that will be inserted are shown below.

101	Smita Kumar	Secretary	28	39500.00
102	Mani Scott	Programmer	32	45300.00
103	Firdaus Ali	Programmer II	45	67500.00

Write SQL query to create the above table with appropriate data types and sizes of columns.

9. Ms. Rajshri is the Class Teacher of Class XII. She wants to create a table named 'Student' to store marks in different subjects of her class. Identify any 4 columns for the table along with their suitable data types.

10. "XYZ" Company conducts workshops for employees of organizations. The company requires data of workshops that are organized. Write SQL query to create a table 'Workshop' with the following structure:

Field	Type	Constraint
WorkshopId	integer	Primary Key
Title	Varchar(50)	
DateWorkshop	Date	
NumSpeakers	Integer	

11. Ariya wants to add another column 'Gender' in the already existing table 'CUSTOMERS'. She has written the following statement. However, it errors. Rewrite the correct statement.  
**MODIFY TABLE CUSTOMERS GENDER char(1);**
12. Explain the following statement with the help of example:
13. "In a transaction either all the SQL statements be committed or all rolled back."
14. How is HAVING clause similar to WHERE clause? How is HAVING clause different from WHERE clause? Explain with the help of examples of each.
15. Consider the following table 'Transporter' that stores the order details about items to be transported. Write SQL commands for the statements (i) to (viii).

**Table: TRANSPORTER**

ORDERNO	DRIVERNAME	DRIVERGRADE	ITEM	TRAVELDATE	DESTINATION
10012	RAMYADAV	A	TELEVISION	2019-04-19	MUMBAI
10014	SOMNATHSINGH		FURNITURE	2019-01-12	PUNE
10016	MOHANVERMA	B	WASHINGMACH	2019-06-06	LUCKNOW
10018	RISHISINGH	A	REFRIGERATOR	2019-04-07	MUMBAI
10019	RADHEMOHAN		TELEVISION	2019-05-30	UDAIPUR
10020	BISHENPRATAP	B	REFRIGERATOR	2019-05-02	MUMBAI
10021	RAM		TELEVISION	2019-05-03	PUNE

- (i) To display names of drivers and destination city where TELEVISION is being transported.
- (ii) To display driver names and destinations where destination is not MUMBAI.
- (iii) To display the names of destination cities where items are being transported. There should be no duplicate values.
- (iv) To display details of rows that have some value in DRIVERGRADE column.
- (v) To display names of drivers, names of items and travel dates for those items that are being transported on or before 1st April 2019.
- (vi) To display the number of drivers who have 'MOHAN' anywhere in their names.
- (vii) To display the names of drivers, item names and travel dates in alphabetic (ascending) order of driver names.
- (viii) To display names of drivers whose names are three characters long.
17. In CHAR(10) and VARCHAR(10), what does the number 10 indicate ?
18. 'Employee' table has a column named 'CITY' that stores city in which each employee resides. Write SQL query to display details of all rows except those rows that have CITY as 'DELHI' or 'MUMBAI' or 'CHANDIGARH'.
19. Consider the table given below. Write SQL queries for (i) to (vii).

**Table: Gym**

REGID	NAME	PREWEIGHT	CURRWEIGHT	DOJ	GENDER	BRANCH
1001	SHEELASIA	115	98	2017-09-02	F	ADARSHVIHAR
1002	FAWADKHAN	92	80	2018-10-11	M	MODEL TOWN

1003	PAWANSINGH	85	80	2018-02-03	M	NIRMANNAGAR
1004	SUPRIYA ARORA	113	100	2018-01-16	F	MODEL TOWN
1005	AJOYBASU	67	75	2017-12-09	M	NIRMANNAGAR
1006	TANMAYJACOB	100	71	2017-11-18	M	ADARSHVIHAR
1007	LAKSHMIVENKAT	98	95	2018-06-09	F	MODEL TOWN

Columns REGID stores Registration Id, PREWEIGHT stores weight of the person before joining Gym, CURRWEIGHT stores current weight, DOJ stores Date of Joining, BRANCH stores the branch of Gym where the person has enrolled.

- (i) To display names of members along with their previous and current weights who are in Model Town branch.
- (ii) To display all names of members, previous weight, current weight, Change in weight (i.e. how much increase from previous weight or decrease from previous weight, Decrease will be displayed with negative sign)
- (iii) To display BRANCH wise count of members in the Gym. (i.e. display the BRANCH and number of members in each BRANCH)
- (iv) To display names and date of joining of all the members who joined in the year 2018.
- (v) To display Names and Current weight of all the members in descending order of Current Weight.
- (vi) To display the names and date of joining of male members who have joined after 27th September 2018.
- (vii) To display names and date of joining of members who have their names starting with 'S' and ending with 'a'.

20. Consider the table Flight given below, write command in SQL for (i) to (iv) and output for (v) to (viii)

Table : FLIGHT

Flight_No	Origin	Destination	Seats	FlightDate	Rate
1005	Varanasi	Nepal	275	12-Dec-07	3000
2785	Delhi	Kerala	290	17-Jan-08	5500
6587	Mumbai	Varanasi	435	19-Feb-08	5000
1265	Varanasi	Nepal	200	02-Jan-08	5400
4457	Delhi	Lucknow	150	22-Feb-08	4500
6856	Varanasi	Mumbai	180	03-Mar-08	6000

- i) To display Flight flying between Varanasi and Nepal.
- ii) To display the different Origin of Flights.
- iii) To display list of flights in descending order of Rate.
- iv) To display flight details of the flight whose flightdate is after Jan 2008.
- v) `SELECT Flight_No, Destination FROM Flight WHERE Destination LIKE '_u%';`
- vi) `SELECT Origin, COUNT(*) FROM Flight GROUP BY Origin;`
- vii) `SELECT Origin, Destination FROM Flight WHERE seats>400;`
- viii) `SELECT SUM(Rate),MAX( Seats) FROM Flight;`

#### JOINS

- A relational database consists of multiple related tables linking together using common columns, which are known as foreign key columns.
- **It is used retrieve data from multiple tables.**

- Consider the tables below EMP, DEPT & SALGRADE that stored related information, all the examples on join will be explained with help of these following three tables

EMP Table

empno	ename	job	mgr	hiredate	sal	comm	deptno
7369	SMITH	CLERK	7902	1993-06-13	800.00	0.00	20
7499	ALLEN	SALESMAN	7698	1998-08-15	1600.00	300.00	30
7521	WARD	SALESMAN	7698	1996-03-26	1250.00	500.00	30
7566	JONES	MANAGER	7839	1995-10-31	2975.00		20
7698	BLAKE	MANAGER	7839	1992-06-11	2850.00		30
7782	CLARK	MANAGER	7839	1993-05-14	2450.00		10
7788	SCOTT	ANALYST	7566	1996-03-05	3000.00		20
7839	KING	PRESIDENT		1990-06-09	5000.00	0.00	10
7844	TURNER	SALESMAN	7698	1995-06-04	1500.00	0.00	30
7876	ADAMS	CLERK	7788	1999-06-04	1100.00		20
7900	JAMES	CLERK	7698	2000-06-23	950.00		30
7934	MILLER	CLERK	7782	2000-01-21	1300.00		10
7902	FORD	ANALYST	7566	1997-12-05	3000.00		20
7654	MARTIN	SALESMAN	7698	1998-12-05	1250.00	1400.00	30

DEPT Table

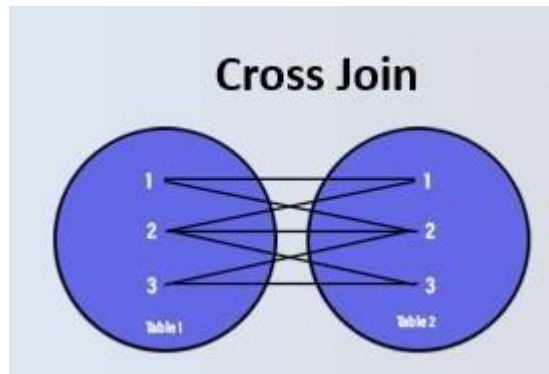
deptno	dname	location
10	Accounting	New York
20	Research	Dallas
30	Sales	Chicago
40	Operations	Boston

SALGRADE Table

grade	losal	hisal
1	700.00	1200.00
2	1201.00	1400.00
4	2001.00	3000.00
5	3001.00	99999.00
3	1401.00	2000.00

- Types of Join
  - ❖ Cartesian Product or Cross join
  - ❖ Equi Join
  - ❖ Natural Join
  - ❖ Non-Equi Join
  - ❖ Self Join
  - ❖ Left Outer Join
  - ❖ Right Outer Join
- Cartesian Product or Cross join
  - ❖ The cross join makes a Cartesian product of rows from the joined tables.

- ❖ The cross join combines each row from the first table with every row from the right table to make the result set.
- ❖ If Table1 has degree d1 and cardinality c1 and table2 has degree d2 and cardinality c2, their Cartesian Product has degree  $d=d1+d2$  and cardinality  $c=c1*c2$ ;



Ven Diagram

Syntax:

SELECT \* FROM table1, table2;

Or

SELECT \* FROM table1 CROSS JOIN table2;

Or

SELECT \* FROM table1 JOIN table2;

e.g. SELECT \* FROM emp, dept;

SELECT \* FROM emp CROSS JOIN dept;

SELECT \* FROM emp JOIN DEPT;

Output:

empno	ename	job	mgr	hiredate	sal	comm	deptno	deptno	dname	loc
7369	SMITH	CLERK	7902	1980-12-17	800.00	NULL	20	10	ACCOUNTING	NEW YORK
7369	SMITH	CLERK	7902	1980-12-17	800.00	NULL	20	20	RESEARCH	DALLAS
7369	SMITH	CLERK	7902	1980-12-17	800.00	NULL	20	30	SALES	CHICAGO
7369	SMITH	CLERK	7902	1980-12-17	800.00	NULL	20	40	OPERATIONS	BOSTON
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30	10	ACCOUNTING	NEW YORK
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30	20	RESEARCH	DALLAS
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30	30	SALES	CHICAGO
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30	40	OPERATIONS	BOSTON
7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30	10	ACCOUNTING	NEW YORK
7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30	20	RESEARCH	DALLAS
7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30	30	SALES	CHICAGO
7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30	40	OPERATIONS	BOSTON
7566	JONES	MANAGER	7839	1981-04-02	2975.00	NULL	20	10	ACCOUNTING	NEW YORK
7566	JONES	MANAGER	7839	1981-04-02	2975.00	NULL	20	20	RESEARCH	DALLAS
7566	JONES	MANAGER	7839	1981-04-02	2975.00	NULL	20	30	SALES	CHICAGO
7566	JONES	MANAGER	7839	1981-04-02	2975.00	NULL	20	40	OPERATIONS	BOSTON
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30	10	ACCOUNTING	NEW YORK
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30	20	RESEARCH	DALLAS
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30	30	SALES	CHICAGO
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30	40	OPERATIONS	BOSTON
7698	BLAKE	MANAGER	7839	1981-05-01	2850.00	NULL	30	10	ACCOUNTING	NEW YORK
7698	BLAKE	MANAGER	7839	1981-05-01	2850.00	NULL	30	20	RESEARCH	DALLAS
7698	BLAKE	MANAGER	7839	1981-05-01	2850.00	NULL	30	30	SALES	CHICAGO
7698	BLAKE	MANAGER	7839	1981-05-01	2850.00	NULL	30	40	OPERATIONS	BOSTON
7782	CLARK	MANAGER	7839	1981-06-09	2450.00	NULL	10	10	ACCOUNTING	NEW YORK
7782	CLARK	MANAGER	7839	1981-06-09	2450.00	NULL	10	20	RESEARCH	DALLAS
7782	CLARK	MANAGER	7839	1981-06-09	2450.00	NULL	10	30	SALES	CHICAGO
7782	CLARK	MANAGER	7839	1981-06-09	2450.00	NULL	10	40	OPERATIONS	BOSTON
7788	SCOTT	ANALYST	7566	1982-12-09	3000.00	NULL	20	10	ACCOUNTING	NEW YORK
7788	SCOTT	ANALYST	7566	1982-12-09	3000.00	NULL	20	20	RESEARCH	DALLAS
7788	SCOTT	ANALYST	7566	1982-12-09	3000.00	NULL	20	30	SALES	CHICAGO
7788	SCOTT	ANALYST	7566	1982-12-09	3000.00	NULL	20	40	OPERATIONS	BOSTON
7839	KING	PRESIDENT	NULL	1981-11-17	5000.00	NULL	10	10	ACCOUNTING	NEW YORK



```

| 7839|KING |PRESIDENT|NULL|1981-11-17|5000.00| NULL| 10| 20|RESEARCH|DALLAS |
| 7839|KING |PRESIDENT|NULL|1981-11-17|5000.00| NULL| 10| 30|SALES |CHICAGO|
| 7839|KING |PRESIDENT|NULL|1981-11-17|5000.00| NULL| 10| 40|OPERATIONS|BOSTON |
| 7844|TURNER|SALESMAN |7698|1981-09-08|1500.00| 0.00| 30| 10|ACCOUNTING|NEW YORK|
| 7844|TURNER|SALESMAN |7698|1981-09-08|1500.00| 0.00| 30| 20|RESEARCH |DALLAS |
| 7844|TURNER|SALESMAN |7698|1981-09-08|1500.00| 0.00| 30| 30|SALES |CHICAGO |
| 7844|TURNER|SALESMAN |7698|1981-09-08|1500.00| 0.00| 30| 40|OPERATIONS|BOSTON |
| 7876|ADAMS |CLERK |7788|1983-01-12|1100.00| NULL| 20| 10|ACCOUNTING|NEW YORK|
| 7876|ADAMS |CLERK |7788|1983-01-12|1100.00| NULL| 20| 20|RESEARCH |DALLAS |
| 7876|ADAMS |CLERK |7788|1983-01-12|1100.00| NULL| 20| 30|SALES |CHICAGO |
| 7876|ADAMS |CLERK |7788|1983-01-12|1100.00| NULL| 20| 40|OPERATIONS|BOSTON |
| 7900|JAMES |CLERK |7698|1981-12-03|950.00| NULL| 30| 10|ACCOUNTING|NEW YORK|
| 7900|JAMES |CLERK |7698|1981-12-03|950.00| NULL| 30| 20|RESEARCH |DALLAS |
| 7900|JAMES |CLERK |7698|1981-12-03|950.00| NULL| 30| 30|SALES |CHICAGO |
| 7900|JAMES |CLERK |7698|1981-12-03|950.00| NULL| 30| 40|OPERATIONS|BOSTON |
| 7902|FORD |ANALYST |7566|1981-12-03|3000.00| NULL| 20| 10|ACCOUNTING|NEW YORK|
| 7902|FORD |ANALYST |7566|1981-12-03|3000.00| NULL| 20| 20|RESEARCH |DALLAS |
| 7902|FORD |ANALYST |7566|1981-12-03|3000.00| NULL| 20| 30|SALES |CHICAGO |
| 7902|FORD |ANALYST |7566|1981-12-03|3000.00| NULL| 20| 40|OPERATIONS|BOSTON |
| 7934|MILLER|CLERK |7782|1982-01-23|1300.00| NULL| 10| 10|ACCOUNTING|NEW YORK|
| 7934|MILLER|CLERK |7782|1982-01-23|1300.00| NULL| 10| 20|RESEARCH |DALLAS |
| 7934|MILLER|CLERK |7782|1982-01-23|1300.00| NULL| 10| 30|SALES |CHICAGO |
| 7934|MILLER|CLERK |7782|1982-01-23|1300.00| NULL| 10| 40|OPERATIONS|BOSTON |
+___+___+.....+.....+.....+.....+.....+.....+

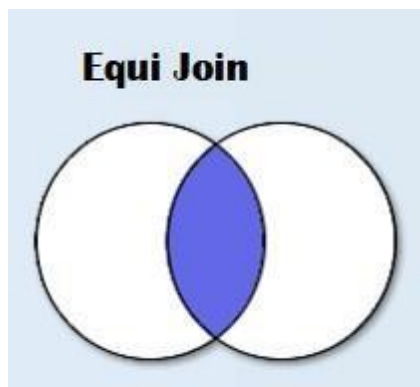
```

56 rows in set (0.02 sec)

Note: **Above output has 56 row (14\*4) and 11 columns (8+3)**

➤ **Equi Join-**

- ❖ It performs a JOIN against equality or matching column(s) values of the associated tables.



Ven Diagram

**Syntax**

```

SELECT */Column_list
FROM Table1, Table 2
WHERE table1.column=Table2.column;

```

Or

```

SELECT */Column_list
FROM Table1 join Table2 on Table1.Column=Table2.Column;

```

Example: `SELECT * FROM emp JOIN dept ON emp.deptno=dept.deptno;`

Or

```

SELECT * FROM emp,dept WHERE emp.deptno=dept.deptno;

```

**Output:**

```

+___+___+.....+.....+.....+.....+.....+.....+
| empno|ename |job |mgr |hiredate |sal |comm |deptno|deptno|dname |loc |
+___+___+.....+.....+.....+.....+.....+.....+
| 7369|SMITH |CLERK |7902|1980-12-17|800.00| NULL| 20| 20|RESEARCH |DALLAS |
| 7499|ALLEN |SALESMAN |7698|1981-02-20|1600.00| 300.00| 30| 30|SALES |CHICAGO |
| 7521|WARD |SALESMAN |7698|1981-02-22|1250.00| 500.00| 30| 30|SALES |CHICAGO |
| 7566|JONES |MANAGER |7839|1981-04-02|2975.00| NULL| 20| 20|RESEARCH |DALLAS |
| 7654|MARTIN|SALESMAN |7698|1981-09-28|1250.00|1400.00| 30| 30|SALES |CHICAGO |
| 7698|BLAKE |MANAGER |7839|1981-05-01|2850.00| NULL| 30| 30|SALES |CHICAGO |

```

```

| 7782|CLARK |MANAGER |7839|1981-06-09|2450.00| NULL| 10| 10|ACCOUNTING|NEW
YORK|
| 7788|SCOTT |ANALYST |7566|1982-12-09|3000.00| NULL| 20| 20|RESEARCH |DALLAS |
| 7839|KING |PRESIDENT|NULL|1981-11-17|5000.00| NULL| 10| 10|ACCOUNTING|NEW
YORK|
| 7844|TURNER|SALESMAN |7698|1981-09-08|1500.00| 0.00| 30| 30|SALES |CHICAGO |
| 7876|ADAMS |CLERK |7788|1983-01-12|1100.00| NULL| 20| 20|RESEARCH |DALLAS |
| 7900|JAMES |CLERK |7698|1981-12-03|950.00| NULL| 30| 30|SALES |CHICAGO |
| 7902|FORD |ANALYST |7566|1981-12-03|3000.00| NULL| 20| 20|RESEARCH |DALLAS |
| 7934|MILLER |CLERK |7782|1982-01-23|1300.00| NULL| 10| 10|ACCOUNTING|NEW YORK|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

Example 1: Display the employee name, sal and name of department name

Ans: In the above query ename and sal belong to emp table whereas dname belongs to DEPT table. So, to retrieve data in this we will use join

```

SELECT emp.ename, emp.sal, dept.dname
FROM emp, dept WHERE emp.deptno=dept.deptno;

```

Output:

```

+-----+-----+-----+
| ename | sal   | dname   |
+-----+-----+-----+
| SMITH | 800.00 | RESEARCH |
| ALLEN | 1600.00 | SALES   |
| WARD  | 1250.00 | SALES   |
| JONES | 2975.00 | RESEARCH |
| MARTIN | 1250.00 | SALES   |
| BLAKE | 2850.00 | SALES   |
| CLARK | 2450.00 | ACCOUNTING |
| SCOTT | 3000.00 | RESEARCH |
| KING | 5000.00 | ACCOUNTING |
| TURNER | 1500.00 | SALES   |
| ADAMS | 1100.00 | RESEARCH |
| JAMES | 950.00 | SALES   |
| FORD | 3000.00 | RESEARCH |
| MILLER | 1300.00 | ACCOUNTING |
+-----+-----+-----+

```

Note:

- ❖ In case of join **full qualified (table\_name.column\_name)** name is used to avoid ambiguity as both table contains common columns as PRIMARY KEY and FOREIGN KEY.
- ❖ Table Alias – Like column alias table alias can be used in case of join as given below.  
SELECT e.ename, e.sal FROM emp e, dept d WHERE emp.deptno=dept.deptno;
- ❖ Here 'e' & 'd' are table alias for EMP & DEPT table respectively.

#### ➤ Non- Equi Join

- ❖ It uses comparison operator instead of the equal sign like >, <, >=, <= along with join condition.

Syntax:

```
SELECT */Column_list
```

```
FROM table1, table2,..
```

```
WHERE table1.column relational_operator table2.column;
```

Note: Where relational operator are other than equality operator and between.

Example: Display the employee name, salary and grade of each employee.

Ans: In the above query the involved table are EMP and SALGRADE.

```
SELECT e.ename, e.sal, s.grade
FROM emp e, salgrade s
WHERE e.sal BETWEEN s.losal AND s.hisal;
```

Output:

```
+-----+-----+-----+
|ename | sal   | grade |
+-----+-----+-----+
|SMITH | 800.00 | 1 |
|ALLEN | 1600.00 | 3 |
|WARD  | 1250.00 | 2 |
|JONES | 2975.00 | 4 |
|MARTIN| 1250.00 | 2 |
|BLAKE | 2850.00 | 4 |
|CLARK | 2450.00 | 4 |
|SCOTT | 3000.00 | 4 |
|KING  | 5000.00 | 5 |
|TURNER| 1500.00 | 3 |
|ADAMS | 1100.00 | 1 |
|JAMES | 950.00  | 1 |
|FORD  | 3000.00 | 4 |
|MILLER| 1300.00 | 2 |
+-----+-----+-----+
```

➤ Natural Join

- ❖ A natural join is a type of join operation that creates an implicit join by combining tables based on columns with the same name and data type.
- ❖ It makes the SELECT query simpler with minimal use of conditions.
- ❖ There is no need to specify the name of common column in the SELECT statement.
- ❖ Common column is present only once in the output.

Syntax:

```
SELECT */column_list
FROM Table1 NATURAL JOIN TABLE2;
```

Or

```
SELECT */column_list
FROM TABLE1 join on common_column_name;
```

Example :

```
SELECT * FROM emp JOIN dept USING(deptno);
```

Or

```
SELECT * FROM emp NATURAL JOIN DEPT;
```

Output:

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|deptno|empno|ename|job   |mgr |hiredate |sal  |comm  |dname  |loc  |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 20 | 7369 |SMITH|CLERK |    |1980-12-17| 800.00 | NULL |RESEARCH |DALLAS |
| 30 | 7499 |ALLEN |SALESMAN |7698 |1981-02-20| 1600.00 | 300.00 |SALES  |CHICAGO |
| 30 | 7521 |WARD  |SALESMAN |7698 |1981-02-22| 1250.00 | 500.00 |SALES  |CHICAGO |
| 20 | 7566 |JONES |MANAGER |7839 |1981-04-02| 2975.00 | NULL  |RESEARCH |DALLAS |
| 30 | 7654 |MARTIN|SALESMAN |7698 |1981-09-28| 1250.00 | 1400.00 |SALES  |CHICAGO |
| 30 | 7698 |BLAKE |MANAGER |7839 |1981-05-01| 2850.00 | NULL  |SALES  |CHICAGO |
| 10 | 7782 |CLARK |MANAGER |7839 |1981-06-09| 2450.00 | NULL  |ACCOUNTING |NEW YORK |
| 20 | 7788 |SCOTT |ANALYST |7566 |1982-12-09| 3000.00 | NULL  |RESEARCH |DALLAS |
| 10 | 7839 |KING  |PRESIDENT |NULL |1981-11-17| 5000.00 | NULL  |ACCOUNTING |NEW YORK |
```

```

| 30 | 7844 | TURNER | SALESMAN | 7698 | 1981-09-08 | 1500.00 | 0.00 | SALES | CHICAGO |
| 20 | 7876 | ADAMS | CLERK | 7788 | 1983-01-12 | 1100.00 | NULL | RESEARCH | DALLAS |
| 30 | 7900 | JAMES | CLERK | 7698 | 1981-12-03 | 950.00 | NULL | SALES | CHICAGO |
| 20 | 7902 | FORD | ANALYST | 7566 | 1981-12-03 | 3000.00 | NULL | RESEARCH | DALLAS |
| 10 | 7934 | MILLER | CLERK | 7782 | 1982-01-23 | 1300.00 | NULL | ACCOUNTING | NEW YORK
|
+_____+_____+_____+_____+_____+_____+_____+_____+_____+_____+

```

➤ Difference between Equi-Join vs Natural Join

Equi-Join	Natural Join
Join performed on equality of value of the columns	Join is performed on column having common name.
Where clause is used to specify the condition	There is no need to use where clause
Both columns from tables are displayed in the result.	Common column is displayed only once

**Solved Exercise on Join:**

Consider the following two table Customer and Saleman

Customer Table:

salesman_id	name	city	commission
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5005	Pit Alex	London	0.11
5006	Mc Lyon	Paris	0.14
5007	Paul Adam	Rome	0.13

Customer Table:

customer_id	cust_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001
3007	Brad Davis	New York	200	5001
3005	Graham Zusi	California	200	5002
3008	Julian Green	London	300	5002
3004	Fabian Johnson	Paris	300	5006
3009	Geoff Cameron	Berlin	100	5003
3003	Jozy Altidor	Moscow	200	5007

Q1. Write a SQL query to display Salesman, cust\_name and city from above table where the salesperson and customer belongs to same city.

Ans:

```

SELECT s.name AS "Salesman",
       c.cust_name, c.city
FROM salesman s, customer c
WHERE s.city=c.city;
Or
SELECT salesman.name AS "Salesman",
       customer.cust_name, customer.city
FROM salesman, customer
WHERE salesman.city=customer.city;

```

Q2. write a SQL query to display ord\_no, purch\_amt, cust\_name, city of those orders where order amount exists between 500 and 2000.

Ans:

```
SELECT o.ord_no,o.purch_amt,
       c.cust_name,c.city
FROM orders o,customer c
WHERE o.customer_id=c.customer_id
AND o.purch_amt BETWEEN 500 AND 2000;
```

Q3. Write a SQL query to display Customer Name, city, Salesman, commission the all salesperson(s) and their respective the customer(s).

Ans:

```
SELECT c.cust_name AS "Customer Name",
       c.city, s.name AS "Salesman", s.commission
FROM customer c, salesman s
WHERE c.salesman_id=s.salesman_id;
```

Q4. What are Joins in SQL?

Ans. Joins in SQL are the statements or clauses using which we can combine two or more tables, based on some common fields present among the tables.

Q5.Explain the different types of Joins in SQL?

Ans: Some of the major joins in SQL are-

- ❖ Equi Join – Inner join is used to return the records which are having matching values in both the tables.
- ❖ Left Join – Left join is used to concatenate all the rows of the left table and the matching rows in the right table.
- ❖ Right Join-Right join is used to concatenate all the rows of the right table and the matching rows in the left table.
- ❖ Full Join-Full join is used to return all the records of both the tables as long as there is a matching record in either table.
- ❖ Self Join-Self join is a join that is used to join a table to itself. In a self-join, a table is considered as if it were two tables.
- ❖ Cartesian Join-Cartesian join is used to return the number of rows in the first table multiplied by the number of rows in the second table. It is also referred to as cross join.

Q6. What is Natural Join?

Ans: Natural join is used to create an implicit join clause based on the value of common attributes in the two tables. Common attributes are the attributes that have the same name in both tables. Natural join does not need any comparison operator as in the case of equi join.

Q7. What is an Equi Join?

Ans: An Equi Join is a type of join that combines tables based on matching values in the specified columns.

- ❖ The column names do not need to be the same.
- ❖ The resultant table can contain repeated columns.
- ❖ It is possible to perform an equi join on more than two tables.

Q 8. What is the difference between cross join and natural join?

Ans: A cross join produces a cross product or cartesian product of two tables whereas the natural join is based on all the columns having the same name and data types in both the tables.

## Interface of python with an SQL database

### Introduction:

- Every application required data to be stored for future reference to manipulate data. Today every application stores data in database for this purpose.
- For example, in school student detail are saved for many reasons like attendance, fee collections, exams, report card etc.
- Python allows us to connect all types of database - Oracle, SQL Server, MySQL, etc.
- In our syllabus we have to understand how to connect python programs with MySQL.
- Before we connect python program with any database like MySQL we need to build a bridge to connect Python and MySQL.
- To Build this bridge we need a connector called “mysql.connector”, which we have to install first.

### Steps to create a database connectivity in Python application are:

1. Open python.
2. Import the package required for database programming:  
Here we need to import **mysql.connector** package in our python scripts.
3. Open a connection:  
Here we need to establish a connection to MySQL database using **connect()**. The syntax for this is as follows:  

```
<Connection_object> = mysql.connector.connect(host= <host_name>, user=  
<user_name>, passwd =<password> , [database = <database>]).
```

  
Here `host_name` means database server name, generally it is given as “localhost”, `user_name` means user by which we connect with MySQL, generally it is given as “root”, `password` is the password of user, `database` is the name of the database whose data we want to use.  
For example:  

```
import mysql.connector as sqltor  
mycon = sqltor.connect(host ="localhost" , user ="root", passwd = "mypass", database  
="test")
```
4. Create a cursor instance:  
Here we have to create an instance of cursor by using **cursor()**, the syntax is as follows:  

```
<cursorobject> =<connectionobject>.cursor()
```

  
For example:  

```
cursor = mycon.cursor()
```
5. Execute a query:  
Here we use the **execute()** with following syntax.  

```
< cursorobject>.execute(<sql query string>)
```

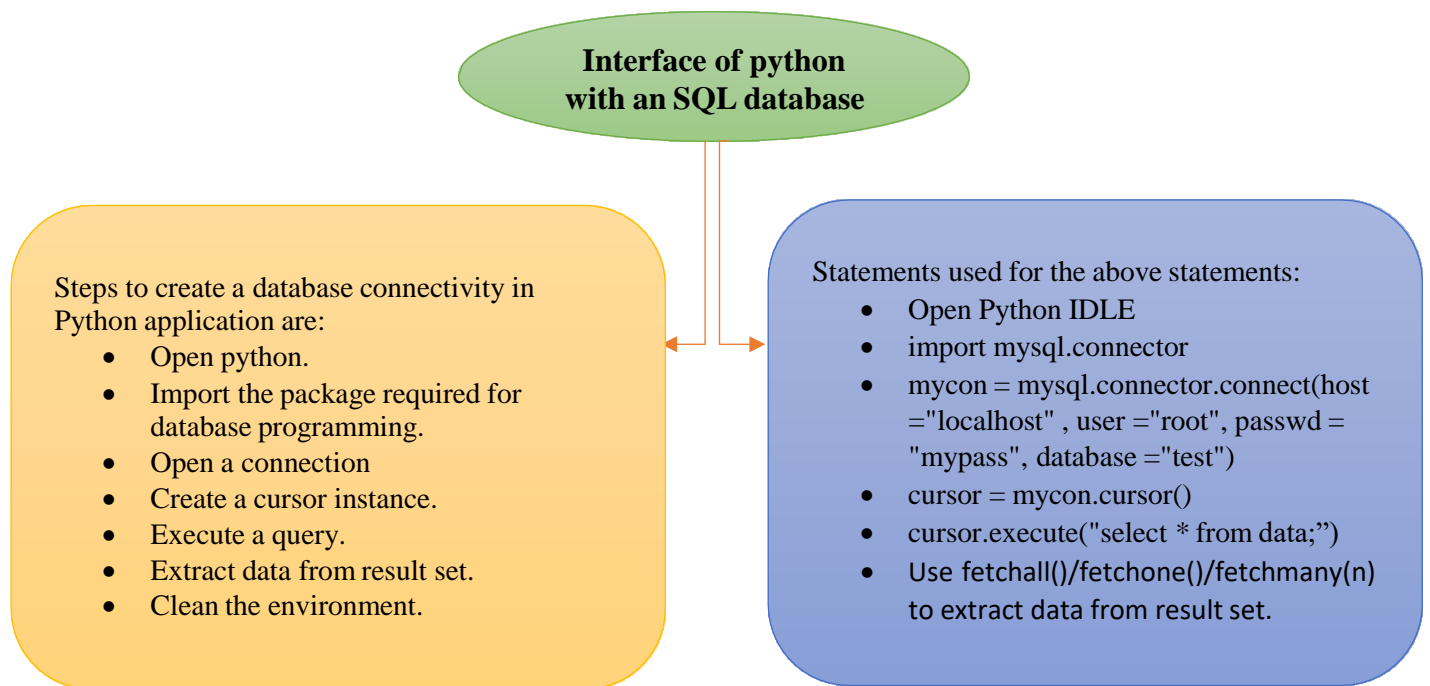
  
For example:  

```
cursor.execute("select * from data;")
```
6. Extract data from result set:  
Here you can fetch the data from the resultset byusing fetch functions.
  - **fetchall():** it will return all th record in the form of tuple.
  - **fetchone():** it returns one record from the result set.
  - **fetchmany(n):** it will return n number of records.
  - **rowcount:** it will return number of rows retrieved from the cursor so far.
7. Clean up the environment.

A database Connection object controls the connection to the database. It represents a unique session with a database connected from within a script/program. A Database Cursor is a special control structure that facilitates the row by row processing of records in the result set. The result set refers to a logical set of records that are fetched from the database by executing an SQL query and made available to the application program.

For INSERT, UPDATE and DELETE queries, we must run **commit()** with the connection object after executing the SQL query.

### Mind Map:



### Questions:

1 Marks Questions	
1	To establish connection between Python and MySQL Which connector is required. (a) mysql.connection      (b) connector      (c) mysql.connect      (d) mysql.connector
2	If you want to check whether connection between Python and MySQL is established or not. Which function is used? (a) con.isconnected()      (b) con.connected()      (c) con.is_connect      (d) con.is_connected
3	Fill the connection string with the required parameter name: - con=mysql.connector.connect(<1>= localhost , user= "scott" <2>= 'tiger' , database='mydb' (a) <1>= host , <2> =passwd      (b) <1>= host , <2> =password (b) <1>= 127.0.0.1 , <2> =password      (d) <1>= 127.0.0.1, <2> =pass
4	Which of the following component act as a container to hold all the data returned from the query and from there we can fetch data at a time? (a) Resultset      (b) Container      (c) Table      (d) Cursor
5	Differentiate between fetchone() and fetchall() ?
6	Which properties of cursor is used to get number of records stored in cursor(Assuming cursor name is dbcursor)? (a) dbcursor.count      (b) dbcursor.row.count (b) dbcursor.records      (d) dbcursor.rowcount
7	Out of which of the following symbol is used for passing parameterized query fro execution to cursor? a) { }      b) %      c) \$      d) Both (a) and (b)
8	Which cursor function is used to fetch n number of records from cursor? a) fetchone()      b) fetch()      c) fetchmany()      d) fetchall()
9	Which cursor function is used to fetch one record at a time from cursor? a) fetchone()      b) Onefetch()      c) fetch()      d) fetchonce()
10	Which cursor function is used to send query to connection? (a) query()      (b) execute()      (c) run()      (d) send()
2 Marks Questions	
1	Consider the following table EMP containing information about an employee: -

EMPNO	ENAME	DEPT	SALARY
1	ALEX	MUSIC	60000
2	PETER	ART	67000
3	JHONY	WE	55000
4	RAMBO	P&HE	48000

Following python code is written to access the records of the table. EMP. What will be the output of the following code:

```
query = "Select * from emp"
mycursor.execute(query)
result1= mycursor.fetchone()
result1= mycursor.fetchone()
result1= mycursor.fetchone()
d= int(result1[3])
print(d*3)
```

- (a) P&HEP&HEP&HE (b) 144000  
(b) WEWEWE (d) 165000

2 Consider the following Python code written to access the record of CODE passes to function. Complete the missing statements:

```
def Search(eno):
    #Assume that all the basic setup import, connection and cursor is already created
    query="select * from emp where emp where empno =_____".format(eno)
    mycursor .execute(query)
    result = mycursor._____
    print(results)
```

- (a) { } and fetchone() (b) fetchone() and { }  
(b) %s and fetchone() (d) %eno and fetchone()

3 Consider the following Python code written for updating the records:

```
def Update(eno):
    #Assume basic setup import, connection(con) and cursor(mycursor) is created
    query = "update emp set salary = 90000 where empno =" +str(eno)
    mycursor.execute(query)
```

Code is running but the record in actual database is not updating, what could be the possible reason?

- (a) save() function is missing (b) con.save() function is missing  
(b) con.commit() function is missing (d) commit() is missing

4 Consider the following python code to display all records from table: EMP

```
def ShowAll()
    #Assume basic setup import, connection(con) and cursor(mycursor) is created
    query="select * from emp"
    mycursor .execute(query)
    result = mycursor. fetchall()
    for results in row:
        print(results)
```

But query is giving error, what could be the possible reason?

- (a) fetchmany() should be used in place of fetchall()  
(b) fetchone() should be used in place of fetchone()  
(c) print(row) should be used in place of print(results)  
(d) loop and print function is wrong, for row in results: snd print(row) should be used.

5 Differentiate between fetchone() and fetchmany() /

6 What is the role of a cursor? Write syntax to create a cursor?

7 To perform query sting based on some parameter. There are two method. Write the given query in (i) String Templates with % formatting and (ii) String Template with { } formatting  
"select all fields from table student where marks >70"



### 4 Marks Questions

1	<p>What is resultset? Write code to execute the query “Select * from student” and store the retrieved record in the cursor object after creating cursor.</p>
2	<p>Explain the purpose of the following lines in the given code:-</p> <pre>import mysql.connection as con mycon = con.connect(host="localhost", user="root",passwd= "admin" , database="employee") mycursor=mycon.cursor()           ←----- 1 mycursor.execute("select * from emp") ← ----- 2 mydata = mycursor.fetchall()      ← -----3 nrec = mycursor.rowcount()        ← -----4 print("Total records fetched till now are:",nrec)</pre>
3	<p>What will be the generated query string of the following code, also write statement to execute Query and save the changes:-</p> <pre>import mysql.connection as con mycon = con.connect(host="localhost", user="root",passwd= "admin" , database="mybooks") mycursor = mycon.cursor() Query = "INSERT INTO books (title, isbn, author, price) VALUES (%s, %s, %s, %s)".%("Panchatantra",123456, "John Fedrik",500.00)</pre>

### 5 Marks Questions

1	<p>Consider the following python and MySQL connectivity code and answer the following questions:</p> <pre>import mysql.connector mydb=mysql.connector.connect(host="localhost",user="root",passwd="12345",database= "student") if mydb.is_connected()==True:     print("connection ok") else:     print("error connecting to mysql database") mycursor=mydb.cursor() r=int(input("enter the rollno")) n=input("enter name") m=int(input("enter marks")) mycursor.execute("INSERT INTO student(rollno,name,marks) VALUES( {},{},{})".format(r,n,m)) mydb.commit() print(mycursor.rowcount,"RECRD INSERTED")</pre> <p>(i) Which of the following statement is connecting database server?  (ii) What is the role of the statement ‘mycursor=mydb.cursor()’?  (iii) Which statement will add record to the table?  (iv) What is the purpose of the ‘cursor.rowcount in the following print statement.  print(mycursor.rowcount,"RECRD INSERTED")  (v) What is the purpose of ‘mydb.commit()’ ?</p>
2	<p>What are the various methods for fetching record from the table? Explain the purpose of different fetch function available in Mysql and fill in the blanks given 1 and 2.</p> <pre>import mysql.connector mydb=mysql.connector.connect(host="localhost",user="root",passwd="12345",database="student") if mydb.is_connected()==True:     print("connection ok") else:     print("error connecting to mysql database") _____1_____ mycursor.execute("SELECT * FROM STUDENT") _____2_____ for row in rs:</pre>

	print(row)
3	<p>Consider the following menu driven program of python and MySQL Connectivity and fill the gap with the correct statement.</p> <pre> import sys mydb=mysql.connector.connect(host="localhost",user="root",passwd="12345") mycursor=mydb.cursor() while True:     print("MENU")     print("1-Creating Database\n 2-Displaying Database List\n 3-Creating Table\n 4- Showing tables List\n 5-To See the structure of Table\n 6-exit\n")     ch=int(input("enter your choice"))     if ch==1:         _____(i)_____     if ch==2:         _____(ii)_____         for x in mycursor:             print (x)     if ch==3:         _____(iii)_____     if ch==4:         _____(iv)_____         for x in mycursor:             print(x)     if ch==5:         _____(v)_____         for x in mycursor:             print(x) sys.exit() </pre> <p>(i) Write command to create database SCHOOL.  (ii) Write command for displaying database list.  (iii) Write statement to create table FEES with FIELD [ROLLNO, NAME, AMOUNT] assume datatype from your own.  (iv) Write statement to display tables in current user.  (v) mycursor.execute("DESC STUDENT")</p>

**Solution:**

1 Marks Questions	
1	(d)mysql.connector
2	(d) con.is_connected()
3	(a) <1>= host , <2> =passwd
4	(d) Cursor
5	fetchone() is used to fetch one record at a time. Subsequent fetchone() will fetch next records.If no more records to fetch ,it returns none. fetchall() is used to fetch all the records from cursor in the form of tuple.
6	(d) dbcursor.rowcount
7	Both (a) and (b)
8	(c) fetchmany()
9	(a) fetchone()
10	(b) execute()
2 Marks Questions	
1	(c) 165000
2	(a) { } and fetchone()
3	(a) con.commit() function is missing

4	(d) loop and print function is wrong, it should be “for row in results” and print(row) should be used.
5	(i) fetchone() ->This method retrieves the next row of a query resultset and returns a single sequence or None if no more rows are available. By default it returns tuple. (ii) fetchmany() -> This method fetches the next set of rows of a query resultset and returns a list of tuples. If no more rows are available it returns an empty list.
6	A database cursor is a special control structure that facilitates the row by row processing of records in the resultset, i.e. the set of records retrieved as per query. <cursor object> = <connection-object>.cursor()
7	(i) st = “Select * from student where marks >%s” %(70) (ii) st= “Select * from student where marks >{ }”.format(70)

#### 4 Marks Questions

1	The resultset refers to a logical set of records that are fetched from the database by executing an SQL query and made available to the application program. cursor = con.cursor() cursor.execute(“select * from student”)
2	1. Create an instance(object) of a cursor using cursor() 2. Executing sql query to retrieve all the records of the table emp in the mycursor object. 3. Fetch all the record in the resultset. 4. To count the number of rows/records returned by sql query.
3	INSERT INTO books (title, isbn, author, price) VALUES (“Panchatantra”,123456, “John Fedrik”,500.00) mycursor.execute(Query) mycon.commit()

#### 5 Marks Questions

1	(i) mysql.connector.connect (ii) Creates an instance of a cursor (iii) mycursor.execute(Sql Query) (iv) To count and display the number of record effected by the query. (v) Confirms the changes made by the user to the database.
2	(i) fetchone() ->This method retrieves the next row of a query resultset and returns a single sequence or None if no more rows are available. By default it returns tuple. (ii) fetchmany() -> This method fetches the next set of rows of a query resultset and returns a list of tuples. If no more rows are available it returns an empty list. (iii) fetchall() -> This method will return all the rows from the resultset in the form of a tuple containing the records. (iv) Fill in the blanks 1 ---- mycursor=mydb.cursor() (v) Fill in the blanks 2 -----rs=mycursor.fetchall()
3	(i) mycursor.execute("CREATE DATABASE SCHOOL") (ii) mycursor.execute("SHOW DATABASE") (iii) mycursor.execute("CREATE TABLE FEES (ROLLNO INTEGER(3),NAME VARCHAR(20), AMOUNT INTEGER(10));") (iv) mycursor.execute("SHOW TABLES") (v) Write statement to display the structure of the table.