

केंद्रीय विद्यालय क्षेत्रीय कार्यालय अहमदाबाद

Kendriya Vidyalaya Sangathan Regional Office, Ahmedabad



CLASS XII

Computer Science(083)

Based on Latest CBSE Exam Pattern

Session 2022-2023

OUR PATRON



**SMT SHRUTI BHARGAV
Deputy Commissioner,
KVS RO Ahmedabad**



**SMT VINITA SHARMA
Assistant Commissioner,
KVS RO Ahmedabad**

SUBJECT CO-ORDINATION BY



SMT SUNITA GUPTA
Principal, KV No 1 AFS Jamnagar

CONTENT TEAM

SMT RAKHI JAIN	PGT COMP	KV NO.1, AFS, JAMANGAR
SH SATISH MAMNANI	PGT COMP	KV AFS , SAMANA
SMT RENU BAHETI	PGT COMP	KV SAC, AHMEDABAD
SMT SONIA CHOUDHARY	PGT COMP	KV NO.2 AHMEDABAD CANTT
SH RAHUL AWASTHI	PGT COMP	KV JUNAGADH
SH DHIREN K MESHVANIYA	PGT COMP	KV JETPUR

TERM WISE SPLIT-UP SYLLABUS**SUB: COMPUTER SCIENCE (083)****CLASS - XII (NEW SYLLABUS)****(SESSION 2022 - 23)****DISTRIBUTION OF MARKS**

UNIT	UNIT NAME	THEORY MARKS	PERIODS	
			THEORY	PRACTICAL
1	Computational Thinking and Programming -2	40	70	50
2	Computer Network	10	15	--
3	Data Management	20	25	20
	TOTAL	70	110	70

MONTH- WISE DISTRIBUTION

Month	Topics to be covered	Th.	Pr.
April	Unit I: Computational Thinking and Programming - 2 <ul style="list-style-type: none"> Revision of Python topics covered in Class XI. 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) 	25	20
May-June	<ul style="list-style-type: none"> Introduction to files, types of files (Text file, Binary file, CSV file), relative and absolute paths 	10	5
July	<ul style="list-style-type: none"> Text file: opening a text file, text file open modes (r, r+, w, w+, a, a+), closing a textfile, opening a file using with clause, writing/appending data to a text file using write() and writelines(), reading from a text file using read(), readline() and readlines(), seek and tell methods, manipulation of data in a text file Binary file: basic operations on a binary file: open using file open modes (rb, rb+, wb, wb+, ab, ab+), close a binary file, import pickle module, dump() and load() method, read, write/create, search, append and update operations in a binary file CSV file: import csv module, open / close csv file, write into a csv file using csv.writerow() and read from a csv file using csv.reader() 	20	15
August	<ul style="list-style-type: none"> Data Structure: Stack, operations on stack (push & pop), implementation of stack using list. 	15	10
September	<ul style="list-style-type: none"> Test (Syllabus covered above) 		

	Unit III: Database Management <ul style="list-style-type: none"> • Database concepts: introduction to database concepts and its need • Relational data model: relation, attribute, tuple, domain, degree, cardinality, keys (candidate key, primary key, alternate key, foreign key) Structured Query Language: introduction, Data Definition Language and Data Manipulation Language, data type (char(n), varchar(n), int, float, date), constraints (not null, unique, primary key), create database, use database, show databases, drop database, show tables, create table, describe table, alter table (add and remove an attribute, add and remove primary key), drop table, insert, delete, update 	10	10
October	<ul style="list-style-type: none"> • select, operators (mathematical, relational and logical), aliasing, distinct clause, where clause, in, between, order by, meaning of null, is null, is not null, like, update command, delete command, aggregate functions (max, min, avg, sum, count), group by, having clause, joins: cartesian product on two tables, equi-join and natural join • Interface of python with an SQL database: connecting SQL with Python, performing insert, update, delete queries using cursor, display data by using fetchone(), fetchall(), rowcount, creating database connectivity applications 	15	10
November	<ul style="list-style-type: none"> • Evolution of networking: introduction to computer networks, evolution of networking (ARPANET, NSFNET, INTERNET) • Data communication terminologies: concept of communication, components of data communication (sender, receiver, message, communication media, protocols), measuring capacity of communication media (bandwidth, data transfer rate), IP address, switching techniques (Circuit switching, Packet switching) • Transmission media: Wired communication media (Twisted pair cable, Co-axial cable, Fiber-optic cable), Wireless media (Radio waves, Micro waves, Infrared waves) • Network devices (Modem, Ethernet card, RJ45, Repeater, Hub, Switch, Router, Gateway, WIFI card) • Network topologies and Network types: types of networks (PAN, LAN, MAN, WAN), networking topologies (Bus, Star, Tree) • Network protocol: HTTP, FTP, PPP, SMTP, TCP/IP, POP3, HTTPS, TELNET, VoIP, wireless/mobile communication protocol such as GSM, GPRS and WLL • Mobile telecommunication technologies: 1G, 2G, 3G, 4G and 5G <p>Introduction to web services: WWW, Hyper Text Markup Language (HTML), Extensible Markup Language (XML), domain names, URL, website, web browser, web servers, web hosting</p> <ul style="list-style-type: none"> • Revision, Project Work 	15	0
December	Pre Board Exam		
January February	Revision and practical Exam		-
March	CBSE BOARD EXAM 2022-23		

**GUIDELINES FOR PRACTICAL
WORK**
COMPUTER SCIENCE
(083) CLASS - XII

S.No		Marks (Total 30)	Term-1 (15 Marks)	Term-2 (15 Marks)
1	Lab Test: 1. Python program	8	6	2
	2. 3 SQL Queries based on one/two table(s), 2 output questions based on SQL queries	4	---	4
2	Report file: Minimum 15 Python programs based on Term - 1 Syllabus SQL Queries – Minimum 5 sets using one table / two tables. Minimum 2 programs based on Python - SQL connectivity.	7	4	3
3	Project (using concepts learnt in Classes 11 and 12) : Coding + Viva voce	8	3	5
4	Viva voce	3	2	1

1. Suggested Practical List:

Python Programming

- Read a text file line by line and display each word separated by a #.
- Read a text file and display the number of vowels/consonants/uppercase/lowercase characters in the file.
- Remove all the lines that contain the character 'a' in a file and write it to another file.
- Create a binary file with name and roll number. Search for a given roll number and display the name, if not found display appropriate message.
- Create a binary file with roll number, name and marks. Input a roll number and update the marks.
- Write a random number generator that generates random numbers between 1 and 6 (simulates a dice).
- Create a CSV file by entering user-id and password, read and search the password for given user-id.
- Write a Python program to implement a stack using list.

Database Management

- Create a student table and insert data. Implement the following SQL commands on the student table:
 - ALTER table to add new attributes / modify data type / drop attribute
 - UPDATE table to modify data
 - ORDER By to display data in ascending / descending order
 - DELETE to remove tuple(s)

- GROUP BY and find the min, max, sum, count and average
 - Joining of two tables.
- Similar exercise may be framed for other cases.
- Integrate SQL with Python by importing suitable module.
- Similar exercise may be framed for other cases.
- Integrate SQL with Python by importing suitable module.

Unit I: Computational Thinking and Programming - 2

Revision of Python topics covered in Class XI

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.

Variables:

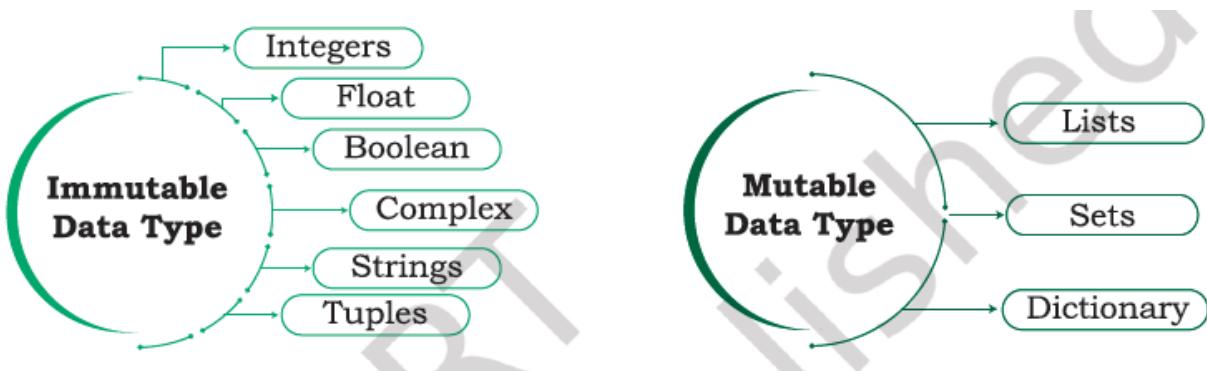
A variable in a program is uniquely identified by a name (identifier). Variable in Python refers to an object — an item or element that is stored in the memory.

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.



• 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 not :

Membership operators : in, not in

• Expressions :

An expression is defined as a combination of constants, variables, and operators. An expression always evaluates to a value. A value or a standalone variable is also considered as an expression but a standalone operator is not an expression.

- | | |
|------------------|---------------------------|
| (i) 100 | (iv) 3.0 + 3.14 |
| (ii) num | (v) 23/3 -5 * 7(14 -2) |
| (iii) num - 20.4 | (vi) "Global" + "Citizen" |

SUMMARY

The if statement is used for selection or decision making.

- The looping constructs while and for allow sections of code to be executed repeatedly under some condition.
- for statement iterates over a range of values or a sequence.
- The statements within the body of for loop are executed till the range of values is exhausted.
- The statements within the body of a while are executed over and over until the condition of the while is false.
- If the condition of the while loop is initially false, the body is not executed even once.
- The statements within the body of the while loop must ensure that the condition eventually becomes false; otherwise, the loop will become an infinite loop, leading to a logical error in the program.
- The break statement immediately exits a loop, skipping the rest of the loop's body. Execution continues with the statement immediately following the body of the loop. When a continue statement is encountered, the control jumps to the beginning of the loop for the next iteration.
- A loop contained within another loop is called a nested loop.

• STRINGS:

Introduction :

String is a sequence which is made up of one or more UNICODE characters. Here the character can be a letter, digit, whitespace or any other symbol. A string can be created by enclosing one or more characters in single, double or triple quote.

```
>>> str1 = 'Hello World!'
>>> str2 = "Hello World!"
>>> str3 = """Hello World!"""
>>> str4 = '"Hello World!"'
```

Indexing :

Each individual character in a string can be accessed using a technique called indexing. The index specifies the character to be accessed in the string and is written in square brackets ([]). The index of the first character (from left) in the string is 0 and the last character is n-1 where n is the length of the string.

Positive Indices	0	1	2	3	4	5	6	7	8	9	10	11
String	H	e	l	l	o		W	o	r	l	d	!
Negative Indices	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

String operations :

(i) Concatenation:

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

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

(ii) Repetition :

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

```
#assign string 'Hello' to str1
>>> str1 = 'Hello'
#repeat the value of str1 2 times
>>> str1 * 2
'HelloHello'
```

(iii) Membership :

Python has two membership operators 'in' and 'not in'. The 'in' operator takes two strings and returns True if the first string appears as a substring in the second string, otherwise it returns False.

```
>>> str1 = 'Hello World!'
>>> 'W' in str1
True
>>> 'Wor' not in str1
False
```

(iv) Slicing :

In Python, to access some part of a string or substring, we use a method called

slicing. This can be done by specifying an index range. Given a string str1, the slice operation str1[n:m] returns the part of the string str1 starting from index n (inclusive) and ending at m (exclusive). In other words, we can say that str1[n:m] returns all the characters starting from str1[n] till str1[m-1]. The numbers of characters in the substring will always be equal to difference of two indices m and n, i.e., (m-n).

```
>>> str1 = 'Hello World!'
#gives substring starting from index 1 to 4
>>> str1[1:5]
'ello'
#gives substring starting from 7 to 9
>>> str1[7:10]
'orl'
#index that is too big is truncated down to
#the end of the string
>>> str1[3:20]
'lo World!'
#first index > second index results in an
#empty " string
>>> str1[7:2]
```

(v) Traversing a string using loops :

We can access each character of a string or traverse a string using for loop and while loop.

(A) String Traversal Using for Loop:

```
>>> str1 = 'Hello World!'
>>> for ch in str1:
print(ch, end = "")
```

Hello World! #output of for loop

In the above code, the loop starts from the first character of the string str1 and automatically ends when the last character is accessed.

(B) String Traversal Using while Loop:

```
>>> str1 = 'Hello World!'
>>> index = 0
#len(): a function to get length of string
>>> while index < len(str1):
print(str1[index],end = "")
index += 1
```

Hello World! #output of while loop

Here while loop runs till the condition index < len(str) is True, where index varies from 0 to len(str1) -1.

(vi) built-in functions:

len(), capitalize(), title(), lower(), upper(), count(), find(), index(), endswith(), startswith(), isalnum(), isalpha(), isdigit(), islower(), isupper(), isspace(), lstrip(), rstrip(), strip(), replace(), join(), partition(), split() :

Method	Description	Example
<code>len()</code>	Returns the length of the given string	<pre>>>> str1 = 'Hello World!' >>> len(str1) 12</pre>
<code>title()</code>	Returns the string with first letter of every word in the string in uppercase and rest in lowercase	<pre>>>> str1 = 'hello WORLD!' >>> str1.title() 'Hello World!'</pre>
<code>lower()</code>	Returns the string with all uppercase letters converted to lowercase	<pre>>>> str1 = 'hello WORLD!' >>> str1.lower() 'hello world!'</pre>
<code>upper()</code>	Returns the string with all lowercase letters converted to uppercase	<pre>>>> str1 = 'hello WORLD!' >>> str1.upper() 'HELLO WORLD!'</pre>
<code>count(str, start, end)</code>	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	<pre>>>> str1 = 'Hello World! Hello Hello' >>> str1.count('Hello',12,25) 2 >>> str1.count('Hello') 3</pre>
<code>find(str,start, end)</code>	Returns the first occurrence of index of substring str occurring 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	<pre>>>> 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</pre>
<code>index(str, start, end)</code>	Same as <code>find()</code> but raises an exception if the substring is not present in the given string	<pre>>>> str1 = 'Hello World! Hello Hello' >>> str1.index('Hello') 0 >>> str1.index('Hee') ValueError: substring not found</pre>
<code>endswith()</code>	Returns True if the given string ends with the supplied substring otherwise returns False	<pre>>>> str1 = 'Hello World!' >>> str1.endswith('World!') True >>> str1.endswith('!') True >>> str1.endswith('lde') False</pre>
<code>startswith()</code>	Returns True if the given string starts with the supplied substring otherwise returns False	<pre>>>> str1 = 'Hello World!' >>> str1.startswith('He') True >>> str1.startswith('Hee') False</pre>

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	<pre>>>> str1 = 'HelloWorld' >>> str1.isalnum() True >>> str1 = 'HelloWorld2' >>> str1.isalnum() True >>> str1 = 'HelloWorld!!' >>> str1.isalnum() False</pre>
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	<pre>>>> str1 = 'hello world!' >>> str1.islower() True >>> str1 = 'hello 1234' >>> str1.islower() True >>> str1 = 'hello ??' >>> str1.islower() True >>> str1 = '1234' >>> str1.islower() False >>> str1 = 'Hello World!' >>> str1.islower() False</pre>
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	<pre>>>> str1 = 'HELLO WORLD!' >>> str1.isupper() True >>> str1 = 'HELLO 1234' >>> str1.isupper() True >>> str1 = 'HELLO ??' >>> str1.isupper() True >>> str1 = '1234' >>> str1.isupper() False >>> str1 = 'Hello World!' >>> str1.isupper() False</pre>

<code>isspace()</code>	Returns True if the string is non-empty and all characters are white spaces (blank, tab, newline, carriage return)	<pre>>>> str1 = '\n \t \r' >>> str1.isspace() True >>> str1 = 'Hello \n' >>> str1.isspace() False</pre>
<code>istitle()</code>	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	<pre>>>> str1 = 'Hello World!' >>> str1.istitle() True >>> str1 = 'hello World!' >>> str1.istitle() False</pre>
<code>lstrip()</code>	Returns the string after removing the spaces only on the left of the string	<pre>>>> str1 = 'Hello World! ' >>> str1.lstrip() 'Hello World!'</pre>
<code>rstrip()</code>	Returns the string after removing the spaces only on the right of the string	<pre>>>> str1 = 'Hello World! ' >>> str1.rstrip() 'Hello World!'</pre>
<code>strip()</code>	Returns the string after removing the spaces both on the left and the right of the string	<pre>>>> str1 = 'Hello World! ' >>> str1.strip() 'Hello World!'</pre>
<code>replace(oldstr, newstr)</code>	Replaces all occurrences of old string with the new string	<pre>>>> str1 = 'Hello World!' >>> str1.replace('o','*') 'Hell* W*rld!' >>> str1 = 'Hello World!' >>> str1.replace('World','Country') 'Hello Country!' >>> str1 = 'Hello World! Hello' >>> str1.replace('Hello','Bye') 'Bye World! Bye'</pre>
<code>join()</code>	Returns a string in which the characters in the string have been joined by a separator	<pre>>>> str1 = ('HelloWorld!') >>> str2 = '-' #separator >>> str2.join(str1) 'H-e-l-l-o-W-o-r-l-d-!'</pre>
<code>partition()</code>	<p>Partitions the given string at the first occurrence of the substring (separator) and returns the string partitioned into three parts.</p> <ol style="list-style-type: none"> 1. Substring before the separator 2. Separator 3. Substring after the separator <p>If the separator is not found in the string, it returns the whole string itself and two empty strings</p>	<pre>>>> str1 = 'India is a Great Country' >>> str1.partition('is') ('India ', 'is', ' a Great Country') >>> str1.partition('are') ('India is a Great Country', ' ', '')</pre>
<code>split()</code>	Returns a list of words delimited by the specified substring. If no delimiter is given then words are separated by space.	<pre>>>> str1 = 'India is a Great Country' >>> str1.split() ['India', 'is', 'a', 'Great', 'Country'] >>> str1 = 'India is a Great Country' >>> str1.split('a') ['Indi', ' is ', ' Gre', ' t Country']</pre>

SUMMARY

A string is a sequence of characters enclosed in single, double or triple quotes.

- Indexing is used for accessing individual characters within a string.
- The first character has the index 0 and the last character has the index n-1 where n is the length

of the string. The negative indexing ranges from -n to -1.

- Strings in Python are immutable, i.e., a string cannot be changed after it is created.
- Membership operator in takes two strings and returns True if the first string appears as a substring in the second else returns False. Membership operator 'not in' does the reverse.
- Retrieving a portion of a string is called slicing. This can be done by specifying an index range.

The slice operation str1[n:m] returns the part of the string str1 starting from index n (inclusive) and ending at m (exclusive).

- Each character of a string can be accessed either using a for loop or while loop.
- There are many built-in functions for working with strings in Python.

• Lists:

Introduction:

The data type list is an ordered sequence which is mutable and made up of one or more elements. Unlike a string which consists of only characters, a list can have elements of different data types, such as integer, float, string, tuple or even another list. A list is very useful to group together elements of mixed data types. Elements of a list are enclosed in square brackets and are separated by comma. Like string indices, list indices also start from 0.

Indexing :

The elements of a list are accessed in the same way as characters are accessed in a string.

List operations (concatenation, repetition, membership & slicing) :

Concatenation

Python allows us to join two or more lists using concatenation operator depicted by the symbol +.

```
#list1 is list of first five odd integers
>>> list1 = [1,3,5,7,9]
#list2 is list of first five even integers
>>> list2 = [2,4,6,8,10]

#elements of list1 followed by list2
>>> list1 + list2
[1, 3, 5, 7, 9, 2, 4, 6, 8, 10]

>>> list3 = ['Red','Green','Blue']
>>> list4 = ['Cyan', 'Magenta', 'Yellow' , 'Black']
>>> list3 + list4
['Red','Green','Blue','Cyan','Magenta', 'Yellow','Black']
```

Repetition

Python allows us to replicate a list using repetition operator depicted by symbol *.

```
>>> list1 = ['Hello']
#elements of list1 repeated 4 times
>>> list1 * 4
['Hello', 'Hello', 'Hello', 'Hello']
```

Membership

Like strings, the membership operators in checks if the element is present in the list and returns True, else returns False.

```
>>> list1 = ['Red','Green','Blue']
>>> 'Green' in list1
True
>>> 'Cyan' in list1
False
```

Slicing

Like strings, the slicing operation can also be applied to lists.

```
>>> list1 =['Red','Green','Blue','Cyan',
'Magenta','Yellow','Black']

>>> list1[2:6]
['Blue', 'Cyan', 'Magenta', 'Yellow']
```

Traversing a list using loops:

We can access each element of the list or traverse a list using a for loop or a while loop.

(A) List Traversal Using for Loop:

```
>>> list1 = ['Red','Green','Blue','Yellow', 'Black']
>>> for item in list1:
    print(item)
```

Output:

Red
Green
Blue
Yellow
Black

built-in functions:

Method	Description	Example
len()	Returns the length of the list passed as the argument	>>> list1 = [10,20,30,40,50] >>> len(list1) 5
list()	Creates an empty list if no argument is passed	>>> list1 = list() >>> list1
	Creates a list if a sequence is passed as an argument	>>> str1 = 'aeiou' >>> list1 = list(str1) >>> list1 ['a', 'e', 'i', 'o', 'u']
append()	Appends a single element passed as an argument at the end of the list	>>> list1 = [10,20,30,40] >>> list1.append(50) >>> list1 [10, 20, 30, 40, 50]
	The single element can also be a list	>>> list1 = [10,20,30,40] >>> list1.append([50,60]) >>> list1 [10, 20, 30, 40, [50, 60]]
extend()	Appends each element of the list passed as argument to the end of the given list	>>> list1 = [10,20,30] >>> list2 = [40,50] >>> list1.extend(list2) >>> list1 [10, 20, 30, 40, 50]
insert()	Inserts an element at a particular index in the list	>>> list1 = [10,20,30,40,50] >>> list1.insert(2,25) >>> list1 [10, 20, 25, 30, 40, 50] >>> list1.insert(0,5) >>> list1 [5, 10, 20, 25, 30, 40, 50]
count()	Returns the number of times a given element appears in the list	>>> list1 = [10,20,30,10,40,10] >>> list1.count(10) 3 >>> list1.count(90) 0
index()	Returns index of the first occurrence of the element in the list. If the element is not present, ValueError is generated	>>> list1 = [10,20,30,20,40,10] >>> list1.index(20) 1 >>> list1.index(90) ValueError: 90 is not in list
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	>>> list1 = [10,20,30,40,50,30] >>> list1.remove(30) >>> list1 [10, 20, 40, 50, 30] >>> list1.remove(90) ValueError: list.remove(x): x not in list
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	>>> list1 = [10,20,30,40,50,60] >>> list1.pop(3) 40 >>> list1 [10, 20, 30, 50, 60] >>> list1 = [10,20,30,40,50,60] >>> list1.pop() 60

reverse()	Reverses the order of elements in the given list	<pre>>>> list1 = [34, 66, 12, 89, 28, 99] >>> list1.reverse() >>> list1 [99, 28, 89, 12, 66, 34] >>> list1 = ['Tiger' , 'Zebra' , 'Lion' , 'Cat' , 'Elephant' , 'Dog'] >>> list1.reverse() >>> list1 ['Dog', 'Elephant', 'Cat', 'Lion', 'Zebra', 'Tiger']</pre>
sort()	Sorts the elements of the given list in-place	<pre>>>> list1=['Tiger','Zebra','Lion', 'Cat', 'Elephant' , 'Dog'] >>> list1.sort() >>> list1 ['Cat', 'Dog', 'Elephant', 'Lion', 'Tiger', 'Zebra'] >>> list1 = [34, 66, 12, 89, 28, 99] >>> list1.sort(reverse = True) >>> list1 [99, 89, 66, 34, 28, 12]</pre>
sorted()	It takes a list as parameter and creates a new list consisting of the same elements arranged in sorted order	<pre>>>> list1 = [23, 45, 11, 67, 85, 56] >>> list2 = sorted(list1) >>> list1 [23, 45, 11, 67, 85, 56] >>> list2 [11, 23, 45, 56, 67, 85]</pre>
min() max() sum()	<p>Returns minimum or smallest element of the list</p> <p>Returns maximum or largest element of the list</p> <p>Returns sum of the elements of the list</p>	<pre>>>> list1 = [34,12,63,39,92,44] >>> min(list1) 12 >>> max(list1) 92 >>> sum(list1) 284</pre>

SUMMARY

Lists are mutable sequences in Python, i.e., we can change the elements of the list.

- Elements of a list are put in square brackets separated by comma.
- A list within a list is called a nested list. List indexing is same as that of strings and starts at 0. Two way indexing allows traversing the list in the forward as well as in the backward direction.
- Operator + concatenates one list to the end of other list.
- Operator * repeats a list by specified number of times.
- Membership operator in tells if an element is present in the list or not and not in does the opposite.
- Slicing is used to extract a part of the list.
- There are many list manipulation functions including: len(), list(), append(), extend(), insert(), count(), find(), remove(), pop(), reverse(), sort(), sorted(), min(), max(), sum().

•Tuples:

Introduction :

A tuple is an ordered sequence of elements of different data types, such as integer, float, string, list or even a tuple. Elements of a tuple are enclosed in parenthesis (round brackets) and are separated by commas. Like list and string, elements of a tuple can be accessed using index values, starting from 0

Indexing :

Elements of a tuple can be accessed in the same way as a list or string using indexing and slicing.

Tuple is Immutable :

Tuple is an immutable data type. It means that the elements of a tuple cannot be changed after it has been created. An attempt to do this would lead to an error.

```
>>> tuple1 = (1,2,3,4,5)
```

```
>>> tuple1[4] = 10
```

```
TypeError: 'tuple' object does not support item assignment
```

Tuple operations :

Concatenation

Python allows us to join tuples using concatenation operator depicted by symbol +. We can also create a new tuple which contains the result of this concatenation operation.

```
>>> tuple1 = (1,3,5,7,9)
```

```
>>> tuple2 = (2,4,6,8,10)
```

```
>>> tuple1 + tuple2
```

```
#concatenates two tuples
```

```
(1, 3, 5, 7, 9, 2, 4, 6, 8, 10)
```

Repetition

Repetition operation is depicted by the symbol *. It is used to repeat elements of a tuple. We can repeat the tuple elements. The repetition operator requires the first operand to be a tuple and the second operand to be an integer only.

```
>>> tuple1 = ('Hello','World')
```

```
>>> tuple1 * 3
```

```
('Hello', 'World', 'Hello', 'World', 'Hello', 'World')
```

Membership

The in operator checks if the element is present in the tuple and returns True, else it returns False.

```
>>> tuple1 = ('Red','Green','Blue')
```

```
>>> 'Green' in tuple1
```

```
True
```

Slicing

Like string and list, slicing can be applied to tuples also.

```
#tuple1 is a tuple
```

```
>>> tuple1 = (10,20,30,40,50,60,70,80)
```

Built-in functions:

`len()`, `tuple()`, `count()`, `index()`, `sorted()`, `min()`, `max()`, `sum()`

Method	Description	Example
<code>len()</code>	Returns the length or the number of elements of the tuple passed as the argument	<code>>>>tuple1 = (10,20,30,40,50)</code> <code>>>> len(tuple1)</code> <code>5</code>
<code>tuple()</code>	Creates an empty tuple if no argument is passed Creates a tuple if a sequence is passed as argument	<code>>>> tuple1 = tuple()</code> <code>>>> tuple1</code> <code>>>> tuple1 = tuple('aeiou')#string</code> <code>>>> tuple1</code> <code>('a', 'e', 'i', 'o', 'u')</code> <code>>>> tuple2 = tuple([1,2,3]) #list</code> <code>>>> tuple2</code> <code>(1, 2, 3)</code> <code>>>> tuple3 = tuple(range(5))</code> <code>>>> tuple3</code> <code>(0, 1, 2, 3, 4)</code>
<code>count()</code>	Returns the number of times the given element appears in the tuple	<code>>>>tuple1=(10,20,30,10,40,10,50)</code> <code>>>> tuple1.count(10)</code> <code>3</code> <code>>>> tuple1.count(90)</code> <code>0</code>
<code>index()</code>	Returns the index of the first occurrence of the element in the given tuple	<code>>>> tuple1 = (10,20,30,40,50)</code> <code>>>> tuple1.index(30)</code> <code>2</code> <code>>>> tuple1.index(90)</code> <code>ValueError: tuple.index(x): x not in tuple</code>
<code>sorted()</code>	Takes elements in the tuple and returns a new sorted list. It should be noted that, <code>sorted()</code> does not make any change to the original tuple	<code>>>>tuple1=("Rama","Heena","Raj", "Mohsin","Aditya")</code> <code>>>> sorted(tuple1)</code> <code>['Aditya', 'Heena', 'Mohsin', 'Raj', 'Rama']</code>
<code>min()</code> <code>max()</code> <code>sum()</code>	Returns minimum or smallest element of the tuple Returns maximum or largest element of the tuple Returns sum of the elements of the tuple	<code>>>>tuple1 = (19,12,56,18,9,87,34)</code> <code>>>> min(tuple1)</code> <code>9</code> <code>>>> max(tuple1)</code> <code>87</code> <code>>>> sum(tuple1)</code> <code>235</code>

• Dictionary:

Introduction :

The data type *dictionary* fall under mapping. It is a mapping between a *set of keys* and a *set of values*. The key-value pair is called an *item*. A key is separated from its value by a colon(:) and consecutive items are separated by commas. Items in dictionaries are unordered, so we may not get back the data in the same order in which we had entered the data initially in the dictionary.

Creating a Dictionary

To create a dictionary, the items entered are separated by commas and enclosed in curly braces. Each item is a key value pair, separated through colon (:). The keys in the dictionary must be unique and should be of any immutable data type, i.e., number, string or tuple. The values can be repeated and can be of any data type.

```
#dict1 is an empty Dictionary created
#curly braces are used for dictionary
>>> dict1 = {}
>>> {}
>>> dict3 = {'Mohan':95,'Ram':89,'Suhel':92, 'Sangeeta':85}
>>> dict3
{'Mohan': 95, 'Ram': 89, 'Suhel': 92, 'Sangeeta': 85}
```

Accessing items in a dictionary using keys :

We have already seen that the items of a sequence (string, list and tuple) are accessed using a technique called indexing. The items of a dictionary are accessed via the keys rather than via their relative positions or indices. Each key serves as the index and maps to a value. The following example shows how a dictionary returns the value corresponding to the given key:

```
>>> dict3 = {'Mohan':95,'Ram':89,'Suhel':92,'Sangeeta':85}
>>> dict3['Ram']          89
>>> dict3['Sangeeta']     85
#the key does not exist
>>> dict3['Shyam']        KeyError: 'Shyam'
```

Mutability of dictionary (adding a new item, modifying an existing item) :

Dictionaries are mutable which implies that the contents of the dictionary can be changed after it has been created.

Adding a new item

We can add a new item to the dictionary as shown in the following example:

```
>>> dict1 = {'Mohan':95,'Ram':89,'Suhel':92,'Sangeeta':85}
>>> dict1['Meena'] = 78
>>> dict1
{'Mohan': 95, 'Ram': 89, 'Suhel': 92, 'Sangeeta': 85, 'Meena': 78}
```

Modifying an Existing Item

The existing dictionary can be modified by just overwriting the key-value pair.

Example to modify a given item in the dictionary:

```
>>> dict1 = {'Mohan':95,'Ram':89,'Suhel':92,'Sangeeta':85}
#Marks of Suhel changed to 93.5
```

```
>>> dict1['Suhel'] = 93.5
>>> dict1
{'Mohan': 95, 'Ram': 89, 'Suhel': 93.5, 'Sangeeta': 85}
```

Membership

The membership operator in checks if the key is present in the dictionary and returns True, else it returns False.

```
>>> dict1 = {'Mohan':95,'Ram':89,'Suhel':92,'Sangeeta':85}
>>> 'Suhel' in dict1
```

True

The not in operator returns True if the key is not present in the dictionary, else it returns False.

```
>>> dict1 = {'Mohan':95,'Ram':89,'Suhel':92,'Sangeeta':85}
>>> 'Suhel' not in dict1
False
```

Traversing a dictionary :

We can access each item of the dictionary or traverse a dictionary using for loop.

```
>>> dict1 = {'Mohan':95,'Ram':89,'Suhel':92, 'Sangeeta':85}
```

Method 1

```
>>> for key in dict1:  
print(key,':',dict1[key])  
Mohan: 95  
Ram: 89  
Suhel: 92  
Sangeeta: 85
```

Method 2

```
>>> for key,value in dict1.items():  
print(key,':',value)  
Mohan: 95  
Ram: 89  
Suhel: 92  
Sangeeta: 85
```

Built-in functions: len(), dict(), keys(), values(), items(), get(), update(), del, clear(), fromkeys(), copy(), pop(), popitem(), setdefault(), max(), min(), count(), sorted(), copy();

Method	Description	Example
len()	Returns the length or number of key: value pairs of the dictionary passed as the argument	<pre>>>> dict1 = {'Mohan':95,'Ram':89,'Suhel':92, 'Sangeeta':85} >>> len(dict1) 4</pre>
dict()	Creates a dictionary from a sequence of key-value pairs	<pre>pair1 = [('Mohan',95),('Ram',89), ('Suhel',92),('Sangeeta',85)] >>> pair1 [('Mohan', 95), ('Ram', 89), ('Suhel', 92), ('Sangeeta', 85)] >>> dict1 = dict(pair1) >>> dict1 {'Mohan':95, 'Ram':89,'Suhel':92,'Sangeeta': 85}</pre>
keys()	Returns a list of keys in the dictionary	<pre>>>> dict1 = {'Mohan':95, 'Ram':89, 'Suhel':92, 'Sangeeta':85} >>> dict1.keys() dict_keys(['Mohan', 'Ram', 'Suhel', 'Sangeeta'])</pre>
values()	Returns a list of values in the dictionary	<pre>>>> dict1 = {'Mohan':95, 'Ram':89, 'Suhel':92, 'Sangeeta':85} >>> dict1.values() dict_values([95, 89, 92, 85])</pre>
items()	Returns a list of tuples(key - value) pair	<pre>>>> dict1 = {'Mohan':95, 'Ram':89, 'Suhel':92, 'Sangeeta':85} >>> dict1.items() dict_items([('Mohan', 95), ('Ram', 89), ('Suhel', 92), ('Sangeeta', 85)])</pre>

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	>>> dict1 = {'Mohan':95, 'Ram':89, 'Suhel':92, 'Sangeeta':85} >>> dict1.get('Sangeeta') 85 >>> dict1.get('Sohan')
update()	appends the key-value pair of the dictionary passed as the argument to the key-value pair of the given dictionary	>>> dict1 = {'Mohan':95, 'Ram':89, 'Suhel':92, 'Sangeeta':85} >>> dict2 = {'Sohan':79,'Geeta':89} >>> dict1.update(dict2) >>> dict1 {'Mohan': 95, 'Ram': 89, 'Suhel': 92, 'Sangeeta': 85, 'Sohan': 79, 'Geeta':89} >>> dict2{'Sohan': 79, 'Geeta': 89}
del()	Deletes the item with the given key To delete the dictionary from the memory we write: del Dict_name	>>> dict1 = {'Mohan':95,'Ram':89, 'Suhel':92, 'Sangeeta':85} >>> del dict1['Ram'] >>> dict1 {'Mohan':95,'Suhel':92, 'Sangeeta': 85} >>> del dict1 ['Mohan'] >>> dict1{'Suhel': 92, 'Sangeeta': 85} >>> del dict1 >>> dict1NameError: name 'dict1' is not defined
clear()	Deletes or clear all the items of the dictionary	>>> dict1 = {'Mohan':95,'Ram':89, 'Suhel':92, 'Sangeeta':85} >>> dict1.clear() >>> dict1 { }

SUMMARY

Tuples are immutable sequences, i.e., we cannot change the elements of a tuple once it is created.

- Elements of a tuple are put in round brackets separated by commas.
- If a sequence has comma separated elements without parentheses, it is also treated as a tuple.
- Tuples are ordered sequences as each element has a fixed position.
- Indexing is used to access the elements of the tuple; two way indexing holds in dictionaries as in strings and lists.
- Operator '+' adds one sequence (string, list, tuple) to the end of other.
- Operator '*' repeats a sequence (string, list, tuple) by specified number of times
- Membership operator 'in' tells if an element is present in the sequence or not and 'not in' does the opposite.
- Tuple manipulation functions are: len(), tuple(), count(), index(), sorted(), min(), max(),sum().
- Dictionary is a mapping (non-scalar) data type. It is an unordered collection of key-value pair; keyvalue pair are put inside curly braces.
- Each key is separated from its value by a colon.
- Keys are unique and act as the index.
- Keys are of immutable type but values can be mutable.

MARKS QUESTIONS

1. Which of the following is valid arithmetic operator in Python: 1
 (i) // (ii) ? (iii) <(iv) and
2. Write the type of tokens from the following: 1
 (i) if (ii) roll_no
3. Name the Python Library modules which need to be imported to invoke the following functions: 1
 (i) sin() (ii) randint ()
4. What do you understand by the term Iteration? 1
5. Which is the correct form of declaration of dictionary? 1
 (i) Day={1:'monday',2:'tuesday',3:'wednesday'}
 (ii) Day=(1;'monday',2;'tuesday',3;'wednesday')
 (iii) Day=[1:'monday',2:'tuesday',3:'wednesday']
 (iv) Day={1'monday',2'tuesday',3'wednesday'}
6. Identify the valid declaration of L: 1
 L = [1, 23, 'hi', 6].
 (i) list (ii) dictionary (iii) array (iv) tuple
7. Find and write the output of the following python code: 1
 x = "abcdef"
 i = "a"
 while i in x:
 print(i, end = " ")
8. Find and write the output of the following python code: 1
 a=10
 def call():
 global a
 a=15
 b=20
 print(a)
 call()
9. Find the valid identifier from the following 1
 a) My-Name b) True c) 2ndName d) S_name
10. Given the lists L=[1,3,6,82,5,7,11,92] , 1
 What will be the output of print(L[2:5])
11. Write the full form of IDLE. 1
12. Identify the valid logical operator in Python from the following. 1
 a) ? b) < c) ** d) and
13. Suppose a tuple Tup is declared as Tup = (12, 15, 63, 80), 1
 which of the following is incorrect?
 a) print(Tup[1])
 b) Tup[2] = 90
 c) print(min(Tup))
 d) print(len(Tup))

- 14 Write a statement in Python to declare a dictionary whose keys are 1,2,3 and values are Apple, Mango and Banana respectively. 1
- 15 A tuple is declared as T = (2,5,6,9,8) 1
What will be the value of sum(T)?
- 16 Name the built-in mathematical function / method that is used to return square root of a number. 1
- 17 If the following code is executed, what will be the output of the following code? str="KendriyaVidyalayaSangathan" 1
print(str[8:16])
- 18 Which of the following are valid operators in Python: (i) ** (ii) between (iii) like (iv) || 1
- 19 Given the lists L=["H", "T", "W", "P", "N"] , write the output of print(L[3:4]) 1
- 20 What will be the output of: print(10>20) 1

2 MARKS QUESTIONS

1. Rewrite the following code in python after removing all syntax error(s). Underline each correction done in the code. 2
30=To
for K in range(0,To)
 IF k%4==0:
 print (K*4)
 Else:
 print (K+3)
2. Find and write the output of the following python code: 2
def fun(s):
 k=len(s)
 m=" "
 for i in range(0,k):
 if(s[i].isupper()):
 m=m+s[i].lower()
 elif s[i].isalpha():
 m=m+s[i].upper()
 else:
 m=m+'bb'
 print(m)
- fun('school2@com')
3. What possible outputs(s) are expected to be displayed on screen at the time of execution of the program from the following code? Also specify the maximum values that can be assigned to each of the variables FROM and TO. 2
import random
AR=[20,30,40,50,60,70];
FROM=random.randint(1,3)
TO=random.randint(2,4)
for K in range(FROM,TO+1):
 print (AR[K],end="# ")
- (i) 10#40#70# (ii) 30#40#50#

- | | | | |
|---|---|----------------|---|
| | (iii) 50#60#70# | (iv) 40#50#70# | |
| 4 | What do you understand by local and global scope of variables? How can you access a global variable inside the function, if function has a variable with same name. | | 2 |
| 5 | Evaluate the following expressions: | | 2 |
| | a) $8 * 3 + 2^{**3} // 9 - 4$ | | |
| | b) $12 > 15$ and $8 > 12$ or not $19 > 4$ | | |
| 6 | Differentiate between call by value and call by reference with a suitable example for each. | | 2 |
| 7 | Rewrite the following code in Python after removing all syntax error(s). Underline each correction done in the code. | | 2 |

```
p=30  
for c in range(0,p)  
If c%4==0:  
    print (c*4)  
Elseif c%5==0:  
    print (c+3)  
else  
    print(c+10)
```

- 8 What possible outputs(s) are expected to be displayed on screen at the time of execution of the program from the following code? Also specify the maximum values that can be assigned to each of the variables Lower and Upper. 2

```
import random  
AR=[20,30,40,50,60,70];  
Lower =random.randint(1,4)  
Upper =random.randint(2,5)  
for K in range(Lower, Upper +1):  
    print (AR[K],end="#")
```

- (i) 40# (ii) 40#50#60# (iii) 50# (iv) All of these

- 9 Evaluate the following expression. 2

 - a) $51+4-3^{**}3//19-3$
 - b) $17 < 19 \text{ or } 30 > 18 \text{ and not } 19 == 0$

- 10 What do you mean by keyword argument in python? Describe with example. 2

- 11 Rewrite the following code in python after removing all syntax errors. 2
Underline each correction done in the code:

```
def func(a):\n    for i in (0,a):\n        if i%2 ==0:\n            s=s+1
```

```

else if i%5==0
    m=m+2
else:
    n=n+i
    print(s,m,n)
func(15)

```

- 12 What possible outputs(s) are expected to be displayed on screen at the time of execution of the program from the following code. Select which option/s is/are correct 2

```

import random

print(random.randint(15,25) , end=' ')
print((100) + random.randint(15,25) , end = ' ')
print((100) -random.randint(15,25) , end = ' ')
print((100) *random.randint(15,25) )

```

- (i) 15 122 84 2500
- (ii) 21 120 76 1500
- (iii) 105 107 105 1800
- (iv) 110 105 105 1900

(i) (ii) are correct answers.

- 13 What is Constraint ? Give example of any two constraints. 2
 14 Predict the output of the following code. 2

```

def swap(P ,Q):
    P,Q=Q,P
    print( P,"#",Q)
    return (P)
R=100
S=200
R=swap(R,S)
print(R,"#",S)

```

- 15 Evaluate the following expressions: 2
 a) $6 * 3 + 4**2 // 5 - 8$
 b) $10 > 5$ and $7 > 12$ or not $18 > 3$
 16 Differentiate between actual parameter(s) and a formal parameter(s) with a suitable example for each.
 17 Explain the use of global key word used in a function with the help of a suitable example.
 18 Rewrite the following code in Python after removing all syntax error(s). Underline each correction done in the code.

```

Value=30
for val in range(0,Value)
    If val%4==0:

```

```
    print (val*4)
Elseif val%5==0:
    print (val+3)
Else
```

- 19 What possible outputs(s) are expected to be displayed on screen at the time of execution of the program from the following code? Also specify the maximum values that can be assigned to each of the variables Lower and Upper.

```
import random  
AR=[20,30,40,50,60,70];  
Lower =random.randint(1,3)  
Upper =random.randint(2,4)  
for K in range(Lower, Upper +1):  
    print (AR[K],end="#")
```


- ```
20 def simple_interest(p, r, t):
 return (p*r*t)/100
print (simple_interest(100,5,2))
```

Identify the formal and actual parameters in the above code snippet.  
Define formal and actual parameters in a function.

- ```
21    c = 10
        def add():
            global c
            c = c + 2
            print("Inside add():", c)
```

```
add()  
c=15  
print("In main:", c)
```

```
output:  
Inside add() : 12  
In main: 15
```

Consider the above program and justify the output. What is the output if "global c" is not written in the function add().

- 22 Consider the following function headers. Identify the correct statement and state reason

 - 1) def correct(a=1,b=2,c):
 - 2) def correct(a=1,b,c=3):
 - 3) def correct(a=1,b=2,c=3):
 - 4) def correct(a=1,b,c):

23 What possible outputs(s) are expected to be displayed on screen at the time of execution of the program from the following code? Also specify the maximum AND minimum values that can be assigned to the variable Num when P = 7

```

import random as r
val = 35
P = 7
Num = 0
for i in range(1, 5):
    Num = val + r.randint(0, P - 1)
    print(Num, " $ ", end = "")
    P = P - 1

```

- (a) 41 \$ 38 \$ 38 \$ 37 \$
 (b) 38 \$ 40 \$ 37 \$ 34 \$
 (c) 36 \$ 35 \$ 42 \$ 37 \$
 (d) 40 \$ 37 \$ 39 \$ 35 \$

24 Find the output of the following program;

```

def increment(n):
    n.append([4])
    return n
l=[1,2,3]
m=increment(l)
print(l,m)

```

25 Evaluate the following:

- a. $45 < 89$ and $7 > 12$ or not $18 > 3$
 b. $6 * 3 + 2 ** 4 // 5 - 2$

3 MARKS QUESTIONS

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

def Change(P ,Q=30):
 P=P+Q
 Q=P-Q
 print(P,"#",Q)
 return (P)

```
- R=150  
 S=100  
 R=Change(R,S)  
 print(R,"#",S)  
 S=Change(S)
2. Write a program in Python, which accepts a list Lst of numbers and n is a numeric value by which all elements of the list are shifted to left.

Sample Input Data of the list :

Lst= [ 10,20,30,40,12,11], n=2

Output Lst = [30,40,12,11,10,20]

3. Write a program in Python, which accepts a list Arr of numbers , the function will replace the even number by value 10 and multiply odd number by 5 .

Sample Input Data of the list is:

```
arr=[10,20,23,45]
output : [10, 10, 115, 225]
```

- 4 Write a program in Python, which accepts a list Arr of numbers and n is a numeric value by which all elements of the list are shifted to left.

Sample Input Data of the list :  
Arr= [ 10,20,30,40,12,11], n=2  
Output :  
Arr = [30,40,12,11,10,20]

- 5 Write a program to reverse elements in a list where arguments are start and end index of the list part which is to be reversed. Assume that start<end, start>=0 and end<len(list) 3

Sample Input Data of List :  
my\_list=[1,2,3,4,5,6,7,8,9,10]  
Output is  
my\_list=[1,2,3,7,6,5,4,8,9,10]

- 6 Write program to add those values in the list of NUMBERS, which are odd. 3

Sample Input Data of the List  
NUMBERS=[20,40,10,5,12,11]  
OUTPUT is 16

- 7 Write a program to replaces elements having even values with its half and elements having odd values with twice its value in a list. 3

eg: if the list contains  
3, 4, 5, 16, 9  
then rearranged list as  
6, 2,10,8, 18

- 8 Write a Python program to find the maximum and minimum elements in the list entered by the user. 3

- 9 Write a program in python to print b power p. 3

- 10 Write a program to print of fibonnaci series upto n. 3

for example if n is 50 then output will be :

```
0
1
1
2
3
5
8
13
21
34
```

**MARKING SCHEME  
1 MARKS QUESTIONS**

|        |                                                                                                                                                                                                                                                                  |   |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
| 1.     | Which of the following is valid arithmetic operator in Python:<br>(i) // (ii) ? (iii) <(iv) and                                                                                                                                                                  | 1 |
| Answer | (i) //                                                                                                                                                                                                                                                           |   |
| 2.     | Write the type of tokens from the following:<br>(i) if (ii) roll_no                                                                                                                                                                                              | 1 |
| Answer | (i) Key word (ii) Identifier<br>(1/2 mark for each correct type)                                                                                                                                                                                                 |   |
| 3.     | Name the Python Library modules which need to be imported to invoke the following functions:<br>(i) sin( ) (ii) randint ( )                                                                                                                                      | 1 |
| Answer | (i) math (ii) random<br>(1/2 mark for each module)                                                                                                                                                                                                               |   |
| 4      | What do you understand by the term Iteration?                                                                                                                                                                                                                    | 1 |
| Answer | Repeation of statement/s finite number of times is known as Iteration.<br>(1 mark for correct answer)                                                                                                                                                            |   |
| 5      | Which is the correct form of declaration of dictionary?<br>(i) Day={1:'monday',2:'tuesday',3:'wednesday'}<br>(ii) Day=(1;'monday',2;'tuesday',3;'wednesday')<br>(iii) Day=[1:'monday',2:'tuesday',3:'wednesday']<br>(iv) Day={1'monday',2'tuesday',3'wednesday'} | 1 |
| Answer | (i) Day={1:'monday',2:'tuesday',3:'wednesday'}                                                                                                                                                                                                                   |   |
| 6      | Identify the valid declaration of L:<br>L = [1, 23, 'hi', 6].<br>(i) list (ii) dictionary (iii) array (iv) tuple                                                                                                                                                 | 1 |
| Answer | (i) List                                                                                                                                                                                                                                                         |   |
| 7      | Find and write the output of the following python code:<br>x = "abcdef"<br>i = "a"<br>while i in x:<br>print(i, end = " ")                                                                                                                                       | 1 |
| Answer | aaaaaa----- OR infinite loop                                                                                                                                                                                                                                     |   |
| 8      | Find and write the output of the following python code:<br>a=10<br>def call():<br>global a<br>a=15<br>b=20<br>print(a)<br>call()                                                                                                                                 | 1 |
| Answer | 15                                                                                                                                                                                                                                                               |   |
| 9      | Find the valid identifier from the following<br>a) My-Name b) True c) 2ndName d) S_name                                                                                                                                                                          | 1 |
| Answer | s) S_name                                                                                                                                                                                                                                                        |   |
| 10     | Given the lists L=[1,3,6,82,5,7,11,92] ,                                                                                                                                                                                                                         | 1 |

|        |                                                                                                                                                                                          |   |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
|        | What will be the output of<br>print(L[2:5])                                                                                                                                              |   |
| Answer | [6,82,5]                                                                                                                                                                                 |   |
| 11     | Write the full form of IDLE.                                                                                                                                                             | 1 |
| Answer | Integrated Development Learning Environment                                                                                                                                              |   |
| 12     | Identify the valid logical operator in Python from the following.<br><br>a) ?              b) <              c) **              d) and                                                   | 1 |
| Answer | d) and                                                                                                                                                                                   |   |
| 13     | Suppose a tuple Tup is declared as Tup = (12, 15, 63, 80),<br>which of the following is incorrect?<br><br>a) print(Tup[1])<br>b) Tup[2] = 90<br>c) print(min(Tup))<br>d) print(len(Tup)) | 1 |
| Answer | b) Tup[2]=90                                                                                                                                                                             |   |
| 14     | Write a statement in Python to declare a dictionary whose keys are<br>1,2,3 and values are Apple, Mango and Banana respectively.                                                         | 1 |
| Answer | Dict={1:'Apple', 2: 'Mango',3 : 'Banana'}                                                                                                                                                |   |
| 15     | A tuple is declared as T = (2,5,6,9,8)<br>What will be the value of sum(T)?                                                                                                              | 1 |
| Answer | 30                                                                                                                                                                                       |   |
| 16     | Name the built-in mathematical function / method that is used to<br>return square root of a number.                                                                                      | 1 |
| Answer | sqrt()                                                                                                                                                                                   |   |
| 17     | If the following code is executed, what will be the output of the<br>following code? str="KendriyaVidyalayaSangathan"<br><br>print(str[8:16])                                            | 1 |
| Answer | Vidyalay                                                                                                                                                                                 |   |
| 18     | Which of the following are valid operators in Python:<br><br>(i) **        (ii) between        (iii) like        (iv)                                                                    | 1 |
| Answer | a) (I) and (iv)                                                                                                                                                                          |   |
| 19     | Given the lists L=[“H”, “T”, “W”, “P”, “N”] , write the output of<br>print(L[3:4])                                                                                                       | 1 |

|        |                                             |   |
|--------|---------------------------------------------|---|
| Answer | ["N"]                                       |   |
| 20     | What will be the output of:<br>print(10>20) | 1 |
| Answer | False                                       |   |

## 2 MARKS QUESTIONS

|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |   |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
| 1.     | <p>Rewrite the following code in python after removing all syntax error(s). Underline each correction done in the code.</p> <pre>30=To for K in range(0,To)     IF k%4==0:         print (K*4)     Else:         print (K+3)</pre>                                                                                                                                                                                                                                | 2 |
| Answer | <u>To=30</u> <pre>for K in range(<u>0,To</u>):     <u>if</u> <u>k</u>%4==0:         print (<u>K</u>*4)     <u>else</u>:         print (<u>K</u>+3)</pre> <p>(1/2 mark for each correction)</p>                                                                                                                                                                                                                                                                    |   |
| 2.     | <p>Find and write the output of the following python code:</p> <pre>def fun(s):     k=len(s)     m=" "     for i in range(0,k):         if(s[i].isupper()):             m=m+s[i].lower()         elif s[i].isalpha():             m=m+s[i].upper()         else:             m=m+'bb'     print(m) fun('school2@com')</pre>                                                                                                                                       | 2 |
| Answer | SCHOOLbbbbCOM<br>(2 marks for correct output)<br>Note: Partial marking can also be given                                                                                                                                                                                                                                                                                                                                                                          |   |
| 3.     | <p>What possible outputs(s) are expected to be displayed on screen at the time of execution of the program from the following code? Also specify the maximum values that can be assigned to each of the variables FROM and TO.</p> <pre>import random AR=[20,30,40,50,60,70]; FROM=random.randint(1,3) TO=random.randint(2,4) for K in range(FROM,TO+1):     print (AR[K],end="# ")</pre> <p>(i) 10#40#70# (ii) 30#40#50#<br/> (iii) 50#60#70# (iv) 40#50#70#</p> | 2 |
| Answer | (ii) 30#40#50# Maximum value FROM,TO is 3,4<br>(1/2 mark each for maximum value)<br>(1 mark for correct option)                                                                                                                                                                                                                                                                                                                                                   |   |

|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |   |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
| 4      | What do you understand by local and global scope of variables? How can you access a global variable inside the function, if function has a variable with same name.                                                                                                                                                                                                                                                                                                                                                  | 2 |
| Answer | <p>A global variable is a variable that is accessible globally. A local variable is one that is only accessible to the current scope, such as temporary variables used in a single function definition.</p> <p>A variable declared outside of the function or in global scope is known as global variable.</p> <p>This means, global variable can be accessed inside or outside of the function whereas local variable can be used only inside of the function. We can access by declaring variable as global A.</p> |   |
| 5      | <p>Evaluate the following expressions:</p> <p>a) <math>8 * 3 + 2^{**}3 // 9 - 4</math></p> <p>b) <math>12 &gt; 15</math> and <math>8 &gt; 12</math> or not <math>19 &gt; 4</math></p>                                                                                                                                                                                                                                                                                                                                | 2 |
| Answer | <p>a) 25</p> <p>b) False</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |   |
| 6      | Differentiate between call by value and call by reference with a suitable example for each.                                                                                                                                                                                                                                                                                                                                                                                                                          | 2 |
| Answer | In the event that you pass arguments like whole numbers, strings or tuples to a function, the passing is like call-by-value because you can not change the value of the immutable objects being passed to the function. Whereas passing mutable objects can be considered as call by reference because when their values are changed inside the function, then it will also be reflected outside the function.                                                                                                       |   |
| 7      | <p>Rewrite the following code in Python after removing all syntax error(s). Underline each correction done in the code.</p> <pre>p=30 for c in range(0,p) If c%4==0:     print (c*4) Elseif c%5==0:     print (c+3) else     print(c+10)</pre>                                                                                                                                                                                                                                                                       | 2 |
| Answer | <pre>p=30 for c in range(0,p): <u>if</u> c%4==0:     print (c*4) <u>elif</u> c%5==0:     print (c+3) <u>else</u>:</pre>                                                                                                                                                                                                                                                                                                                                                                                              |   |

|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |   |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
|        | print(c+10)                                                                                                                                                                                                                                                                                                                                                                                                                                                        |   |
| 8      | <p>What possible outputs(s) are expected to be displayed on screen at the time of execution of the program from the following code? Also specify the maximum values that can be assigned to each of the variables Lower and Upper.</p> <pre>import random AR=[20,30,40,50,60,70]; Lower =random.randint(1,4) Upper =random.randint(2,5) for K in range(Lower, Upper +1):     print (AR[K],end="#")</pre> <p>(i) 40# (ii) 40#50#60# (iii) 50# (iv) All of these</p> | 2 |
| Answer | (iv) All of these                                                                                                                                                                                                                                                                                                                                                                                                                                                  |   |
| 9      | <p>Evaluate the following expression.</p> <p>a) <math>51+4-3^{**3}/19-3</math><br/> b) <math>17 &lt; 19</math> or <math>30 &gt; 18</math> and not <math>19 == 0</math></p>                                                                                                                                                                                                                                                                                         | 2 |
| Answer | 51<br>True                                                                                                                                                                                                                                                                                                                                                                                                                                                         |   |
| 10     | What do you mean by keyword argument in python? Describe with example.                                                                                                                                                                                                                                                                                                                                                                                             | 2 |
| Answer | When you assign a value to the parameter (such as param=value) and pass to the function (like fn(param=value)), then it turns into a keyword argument.                                                                                                                                                                                                                                                                                                             |   |
| 11     | <p>Rewrite the following code in python after removing all syntax errors. Underline each correction done in the code:</p> <pre>Def func(a):     for i in (0,a):         if i%2 =0:             s=s+1         else if i%5= =0             m=m+2         else:             n=n+i     print(s,m,n) func(15)</pre>                                                                                                                                                     | 2 |

|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |   |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
| Answer | <pre> def func(a):          #def     s=m=n=0           #local variable     for i in (0,a):   #indentation and range function missing         if i%2==0:             s=s+i         elif i%5==0:    #elif and colon             m=m+i         else:             n=n+i     print(s,m,n)      #indentation func(15) <b>2 marks for any four corrections.</b> </pre>                                                                                                          |   |
| 12     | <p>What possible output(s) are expected to be displayed on screen at the time of execution of the program from the following code. Select which option/s is/are correct</p> <pre> import random print(random.randint(15,25) , end=' ') print((100) + random.randint(15,25) , end = ' ' ) print((100)-random.randint(15,25) , end = ' ' ) print((100)*random.randint(15,25) )  (i) 15 122 84 2500 (ii) 21 120 76 1500 (iii) 105 107 105 1800 (iv) 110 105 105 1900 </pre> | 2 |
| Answer | <u>(i ) (ii) are correct answers.</u>                                                                                                                                                                                                                                                                                                                                                                                                                                    |   |
| 13     | What is Constraint ? Give example of any two constraints.                                                                                                                                                                                                                                                                                                                                                                                                                | 2 |
| Answer | Constraints are the checking condition which we apply on table to ensure the correctness of data . example primary key, not null, default, unique etc                                                                                                                                                                                                                                                                                                                    |   |
| 14     | <p>Predict the output of the following code.</p> <pre> def swap(P ,Q):     P,Q=Q,P     print( P,"#",Q)     return (P) R=100 S=200 R=swap(R,S) </pre>                                                                                                                                                                                                                                                                                                                     | 2 |

|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |   |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
|        | print(R,"#",S)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |   |
| Answer | 200 # 100<br>200 # 200                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |   |
| 15     | Evaluate the following expressions:<br>a) $6 * 3 + 4^{**}2 // 5 - 8$<br>b) $10 > 5 \text{ and } 7 > 12 \text{ or not } 18 > 3$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 2 |
| Answer | a) 13<br>b) False                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |   |
| 16     | Differentiate between actual parameter(s) and a formal parameter(s) with a suitable example for each.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |   |
| Answer | The list of identifiers used in a function call is called actual parameter(s) whereas the list of parameters used in the function definition is called formal parameter(s).<br>Actual parameter may be value / variable or expression.<br>Formal parameter is an identifier.<br>Example:<br><pre>def area(side):<br/>    # line 1<br/>    return side*side;<br/>print(area(5))<br/>    # line 2</pre> In line 1, side is the formal parameter and in line 2, while invoking area() function, the value 5 is the actual parameter.<br>formal parameter, i.e. a parameter, is in the <i>function definition</i> . An actual parameter, i.e. an argument, is in a <i>function call</i> .                              |   |
| 17     | Explain the use of global key word used in a function with the help of a suitable example.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |   |
| Answer | Use of global key word:<br>In Python, global keyword allows the programmer to modify the variable outside the current scope. It is used to create a global variable and make changes to the variable in local context. A variable declared inside a function is by default local and a variable declared outside the function is global by default. The keyword global is written inside the function to use its global value. Outside the function, global keyword has no effect. Example<br><pre>c = 10 # global variable<br/>def add():<br/>    global c<br/>    c = c + 2 # global value of c is incremented by 2<br/>    print("Inside add()", c)</pre><br><pre>add()<br/>c=15<br/>print("In main:", c)</pre> |   |

|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |  |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
|        | <p>output:<br/>Inside add() : 12<br/>In main: 15</p>                                                                                                                                                                                                                                                                                                                                                                                                                                   |  |
| 18     | <p>Rewrite the following code in Python after removing all syntax error(s).<br/>Underline each correction done in the code.</p> <pre>Value=30 for val in range(0,Value)     If val%4==0:         print (val*4)     Elseif val%5==0:         print (val+3)     Else print(val+10)</pre>                                                                                                                                                                                                 |  |
| Answer | <pre>Value=30 for VAL in range(0,Value) :     if val%4==0:         print (VAL*4)     elif val%5==0:         print (VAL+3)     else:         print(VAL+10)</pre>                                                                                                                                                                                                                                                                                                                        |  |
| 19     | <p>What possible outputs(s) are expected to be displayed on screen at the time of execution of the program from the following code? Also specify the maximum values that can be assigned to each of the variables Lower and Upper.</p> <pre>import random AR=[20,30,40,50,60,70]; Lower =random.randint(1,3) Upper =random.randint(2,4) for K in range(Lower, Upper +1): print (AR[K],end="#")</pre> <p>(i) 10#40#70#<br/> (ii) 30#40#50#<br/> (iii) 50#60#70#<br/> (iv) 40#50#70#</p> |  |
| Answer | <p>OUTPUT: (ii)<br/> Maximum value of Lower: 3<br/> Maximum value of Upper: 4</p>                                                                                                                                                                                                                                                                                                                                                                                                      |  |
| 20     | <pre>def simple_interest(p, r, t):     return (p*r*t)/100 print (simple_interest(100,5,2))</pre>                                                                                                                                                                                                                                                                                                                                                                                       |  |



|        |                                                                                                                                              |  |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------|--|
|        | (d) 40 \$ 37 \$ 39 \$ 35 \$<br>Maximum value of Num when P = 7 : 42<br>Minimum value of Num when P = 7 : 35                                  |  |
| 24     | Find the output of the following program;<br><pre>def increment(n):     n.append([4])     return n l=[1,2,3] m=increment(l) print(l,m)</pre> |  |
| Answer | [1, 2, 3, [4]] [1, 2, 3, [4]]                                                                                                                |  |
| 25     | Evaluate the following:<br>a. $45 < 89$ and $7 > 12$ or not $18 > 3$<br>b. $6 * 3 + 2 * * 4 // 5 - 2$                                        |  |
| Answer | a. False<br>b. 19                                                                                                                            |  |

### 3 MARKS QUESTIONS

|        |                                                                                                                                                                                                                                                                   |   |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
| 1.     | Find and write the output of the following python code:<br><pre>def Change(P ,Q=30):     P=P+Q     Q=P-Q     print( P,"#",Q)     return (P)  R=150 S=100 R=Change(R,S) print(R,"#",S) S=Change(S)</pre>                                                           | 3 |
| Answer | 250 # 150<br>250 # 100<br>130 # 100                                                                                                                                                                                                                               |   |
|        | <b>(1 mark each for correct line)</b>                                                                                                                                                                                                                             |   |
| 2.     | Write a program in Python, which accepts a list Lst of numbers and n is a numeric value by which all elements of the list are shifted to left.<br><br>Sample Input Data of the list<br><br>Lst= [ 10,20,30,40,12,11], n=2<br><br>Output Lst = [30,40,12,11,10,20] |   |
| Answer | <pre>L=len(Lst) for x in range(0,n):     y=Lst[0]     for i in range(0,L-1):</pre>                                                                                                                                                                                |   |

|        |                                                                                                                                                                                                                                                                                                                                         |   |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
|        | <pre>Lst[i]=Lst[i+1] Lst[L-1]=y print(Lst) #Note : Using of any correct code giving the same result is also accepted.</pre>                                                                                                                                                                                                             |   |
| 3.     | <p>Write a program in Python, which accepts a list Arr of numbers , the function will replace the even number by value 10 and multiply odd number by 5 .</p> <p>Sample Input Data of the list is:</p> <pre>arr=[10,20,23,45]</pre> <p>output : [10, 10, 115, 225]</p>                                                                   |   |
| Answer | <pre>l=len(arr) for a in range(l):     if(arr[a]%2==0):         arr[a]=10     else:         arr[a]=arr[a]*5 print(arr)</pre> <p>1 mark for function<br/>1 mark for loop and condition checking<br/>1 mark for if and else</p>                                                                                                           |   |
| 4      | <p>Write a program in Python, which accepts a list Arr of numbers and n is a numeric value by which all elements of the list are shifted to left.</p> <p>Sample Input Data of the list</p> <pre>Arr= [ 10,20,30,40,12,11], n=2</pre> <p>Output</p> <pre>Arr = [30,40,12,11,10,20]</pre>                                                 |   |
| Answer | <pre>Arr= [ 10,20,30,40,12,11], n=2</pre> <pre>L=len(Arr) for x in range(0,n):     y=Arr[0]     for i in range(0,L-1):         Arr[i]=Arr[i+1]     Arr[L-1]=y print(Arr)</pre>                                                                                                                                                          |   |
| 5      | <p>Write a program to reverse elements in a list where arguments are start and end index of the list part which is to be reversed. Assume that start&lt;end, start&gt;=0 and end&lt;len(list)</p> <p>Sample Input Data of List</p> <pre>my_list=[1,2,3,4,5,6,7,8,9,10]</pre> <p>Output is</p> <pre>my_list=[1,2,3,7,6,5,4,8,9,10]</pre> | 3 |

|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |   |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |   |
| Answer | <pre>my_mylist=[1,2,3,4,5,6,7,8,9,10]</pre> <p style="text-align: right;"><b>1/2 Mark</b></p> <pre>I1=mylist[:start:1]</pre> <p style="text-align: right;"><b>1/2 Mark</b></p> <pre>I2=mylist[end:start-1:-1]</pre> <p style="text-align: right;"><b>1/2 Mark</b></p> <pre>I3=mylist[end+1:]</pre> <p style="text-align: right;"><b>1/2 Mark</b></p> <pre>final_list=I1+I2+I3</pre> <p style="text-align: right;"><b>1/2 Mark</b></p> <pre>print (final_list)</pre> <p style="text-align: right;"><b>1/2 Mark</b></p> <p>Or any other relevant code</p> |   |
| 6      | <p>Write program to add those values in the list of NUMBERS, which are odd.</p> <p>Sample Input Data of the List</p> <p>NUMBERS=[20,40,10,5,12,11]</p> <p>OUTPUT is 16</p>                                                                                                                                                                                                                                                                                                                                                                              | 3 |
| Answer | <pre>NUMBERS=[20,40,10,5,12,11]</pre> <pre>s=0</pre> <pre>for i in NUMBERS:</pre> <pre>    if i%2 !=0:</pre> <pre>        s=s+i</pre> <pre>print(s)</pre>                                                                                                                                                                                                                                                                                                                                                                                               |   |
| 7      | <p>Write a program to replaces elements having even values with its half and elements having odd values with twice its value in a list.</p> <p>eg: if the list contains<br/>3, 4, 5, 16, 9<br/>then rearranged list as<br/>6, 2,10,8, 18</p>                                                                                                                                                                                                                                                                                                            | 3 |
| Answer | <pre>L=[3, 4, 5, 16, 9]</pre> <pre>for i in range(n):</pre> <pre>    if L[i] % 2 == 0:</pre> <pre>        L[i] /= 2</pre> <pre>    else:</pre> <pre>        L[i] *= 2</pre> <pre>print (L)</pre>                                                                                                                                                                                                                                                                                                                                                        |   |
| 8      | Write a Python program to find the maximum and minimum elements in the list entered by the user.                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 3 |
| Answer | <pre>lst = []</pre> <pre>num = int(input("How many numbers"))</pre> <pre>for n in range (nm):</pre> <pre>    numbers = int(input("Enter number :"))</pre> <pre>    lst.append(numbers)</pre> <pre>print("Maximum element in the list is :",max(lst))</pre> <pre>print("Minimum element in the list is :",min(lst))</pre>                                                                                                                                                                                                                                |   |
| 9      | Write a program in python to print b power p.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 3 |
| Answer | <pre>b=int(input("Enter base number"))</pre> <pre>p=int(input("Enter exp no."))</pre> <pre>t=b</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                    |   |

|        |                                                                                                                                                   |   |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------|---|
|        | <pre>for i in range(p-1):     t=t*b     print(t)</pre>                                                                                            |   |
| 10     | <p>Write a program to print of fibonnaci series upto n.</p> <p>for example if n is 50 then output will be :</p> <pre>0 1 1 2 3 5 8 13 21 34</pre> | 3 |
| Answer | <pre>n=int(input("Enter any number")) a=0 b=0 c=1 while a&lt;=n:     print(a)     b=c     c=a     a=b+c</pre>                                     |   |

## FUNCTIONS

### **Notes**

- The act of partitioning a program into individual components is called "Modularity".
- A module is a separately saved unit whose functionality can be reused.
- A Python module has the .py extension.
- A Python module can contain objects like docstrings, variables, constants, classes, objects, statements, functions etc.
- The Python modules that come preloaded with Python are called "standard library modules".
- A function is a named block of statements that can be invoked by its name.
- Python can have three types of functions i.e., built-in functions, functions in modules and user-defined functions
- The docstrings are useful for documentation purpose.
- Python module can be imported in a program using import statement.
- There are two forms of import statements:
  - (i) import <module name>
  - (ii) from <module name> import <object>
- The built-in functions of Python are always available, one needs not import any module for them.
- The math module of Python provides mathematical functionality.
- sys.stdin is the most widely used method to read input from the command line or terminal.
- The command line sys.argv arguments is another way that we can grab input, and environment be used from within our programs.
- The basic I/O (Input/Output) functions are input() and print() respectively.

- One of the most useful tools available in Python is the `print()` function. This simply allows the program to display or print data for the user to read. For example:  
`red='Hi, how are you?'`

`print(red)`

Output : Hi, how are you?

- Python has an `input` function which lets you ask a user for some text input. In Python 2, you have a built-in function `raw_input()`, whereas in Python 3, you have `input()` for inputting by user. The syntax is:

`mydata = input("Prompt:")`

`print(mydata)`

Output: Prompt:

- In Python, a number mathematical operations can be performed with ease by importing a module named "math" which defines various functions which makes our task easier.

- `ceil(x)`: Returns the ceiling of  $x$  as a float, the smallest integer value greater than or equal to  $x$ .

- `floor(x)`: Returns floor of  $x$  as a float, the largest integer value less than or equal to  $x$ .

- `fabs(x)`: Returns the floating point absolute value of  $x$ .

- `exp(x)`: Return  $e^{**x}$

- `log(x,(base))`: With one argument, returns the natural logarithm of  $x$  (to base e).

With two arguments, returns the logarithm of  $x$  to the given base calculate as  $\log(x)/\log(base)$

- `log10(x)`: Returns the base-10 logarithm of  $x$ . This is usually more accurate than `log(x,10)`.

- `pow(x, y)`: Returns  $x$  raised to the power  $y$ . In particular, `pow(10, x)` and `pow(x, 0.0)` always return 1.0, even when  $x$  is a zero or a NaN. If both  $x$  and  $y$  are finite,  $x$  is negative, and  $y$  is not an integer then `pow(x, y)` is undefined, and raises `ValueError`.

- `sqrt(x)`: Returns the square root of  $x$ .

- `cos(x)`: Returns the cosine of  $x$  radians.

- `sin(x)`: Returns the sine of  $x$  radians.

- `tan(x)`: Returns the tangent of  $x$  radians.

- `degrees(x)`: Converts angle  $x$  from radians to degrees.

- `radians(x)`: Converts angle  $x$  from degrees to radians.

- A function is a block of organized and reusable code that is used to perform a single, related action. Functions provide better modularity for your application and a high degree of code reusability.

- Function blocks begin with the keyword `def` followed by the function name and parentheses `()`.

- Any input parameters or arguments should be placed within these parentheses. You can also define parameters inside these parentheses.

- The first statement of a function can be an optional statement-the documentation string of the function or `docstring`.

- The code block within every function starts with a colon `(:)` and is indented.

- The statement `return [expression]` exit a function, optionally passing back an expression to the caller. A `return` statement with no arguments is the same as `return None`.

- Defining a function only gives a name, specifies the parameters that are to be included in the function, a structure the blocks of code.

- The scope of a variable determines the portion of the program where you can access a particular identifier. There are two basic scopes of variables in Python :

1. Global variables
2. Local variables
  - Variables that are defined inside a function body have a local scope, and those defined outside have a global scope.
- Built-in These are the functions that are always available in Python and can be accessed by a programmer without importing any module.
- Examples of Some Built-in Functions
  - (i) `print()`: It prints objects to the text stream file.
  - (ii) `input()`: It reads the input, converts it to a string and returns that.
  - (iii) `sorted()`: Returns a new sorted list from the items in iterable..
  - (iv) `bool()`: Returns a boolean value i.e., True or False..
  - (v) `min()`: Returns the smallest of two or more arguments.
  - (vi) `any()`: Returns True if any element of the iterable is True.
- String Functions
  - (i) `partition()`: It splits the string at the first occurrence of the given argument and returns a tuple containing three parts.
  - (ii) `join()`: It takes a list of string and joins them as a regular string.
  - (iii) `split()`: It splits the whole string into the items with separator as a delimiter.
- Modules: It is a file containing Python definitions and statements. We need to import modules use any containing part before separator, separator parameter and part after the separator if the separator parameter is found in the string of its function or variable in our code.
- Examples of Some Module Functions
  - (i) `fabs()`: It returns the absolute value of a number.
  - (ii) `factorial()`: This method finds the factorial of a positive integer.
  - (iii) `random()`: It produces an integer between the limit arguments.
  - (iv) `today()`: This method returns the current date and time.
  - (v) `search()`: This function searches the pattern inside the string.
  - (vi) `capitalize()`: It returns the copy of string in capital letters.
- User-Defined Functions: User defined functions are those that we define ourselves in our program and then call them wherever we want.
- Parameters: These are the values provided in the parentheses in the function header when we define the function
- Arguments: These are the values provided in function call/invoke statement.
- Function Arguments: You can call a function by using the following types of formal arguments:
  - Required arguments/Positional arguments
  - Keyword arguments
  - Default arguments
  - Variable-length arguments
- Required arguments are the arguments passed to a function in correct positional order.
- Keyword arguments are related to the function calls. When you use keyword arguments in a function call, the caller identifies the arguments by the parameter name.
- A default argument is an argument that assumes a default value, if a value is not provided in the function call for that argument.
- All variables in a program may not be accessible at all locations in that program. This depends on where you have declared a variable or the scope of variable.

## **Passing different objects as arguments :**

You can send any data types of argument to a function as string, number, list, dictionary etc., and it will be treated as the same data type inside a function.

e.g. List as an argument

```
def fun(Fruit):
 for i in Fruit:
 print(i)
```

```
Food ["Mango", "Cherry", "Grapes", "Banana"]
fun(Food)
```

Output:-

```
Mango
Cherry
Grapes
Banana
```

## **Short Answer Type Questions – (1 mark for correct answer)**

Q.1 Find and write the output of the following python code:

```
a=10
def call():
 global a
 a=15
call()
print(a)
```

Q.2. What do you mean by modularity?

Q.3. What is a function call?

Q.4. Name the three categories of functions.

Q.5. What is the role of an argument of a function?

Q.6. What are docstrings?

Q.7. What are docstring conventions?

Q.8. What is the use of following functions?

Q.9. Name the constant available in math module.

Q.10. Write two ways in which you are able to use constant pi in your programs.

Q.11. What is dot notation?

Q.12. What is a function?

Q.13. What is an argument?

Q.14. What is the general syntax for defining a function in Python?

Q.15. What is function header?

Q.16. What are parameters?

Q.17. The convention for indentation within a block is four spaces. Is it true?

Q.18. Find the error in the following codes.

```
def minus(total_decrement)
 output= total_decrement
```

Q. 19. What is \_main\_?

Q. 20. What is \_name\_?

Q. 21. What is the difference between arguments and parameters?

Q. 22. Name two types of function in Python.

Q. 23. Trace the flow of execution for following program.

```
1 def power (b,p):
2 r = b**p
```

```

3 return r
4
5 def calcsquare(a):
6 a= power(a,2)
7 return a
8
9 n=5
10 result = calcsquare(n)
11 print(result)

```

Q. 24. What will be the output of the following code?

```

def addEm(x,y,z):
 print(x+y+z)

```

```

def prod(x,y,z):
 return x*y*z

```

```

a= addEm(6,16,26)
b= prod(2,3,6)
print(a,b)

```

Q.25. What is the use of return statement?

#### **Short Long Answer Type Questions - (2 mark for correct answer)**

Q. 1. Find and write the output of the following Python code:

```

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

```

Q. 2. What do you understand by local and global scope of variables? How can you access a global variable inside the function, if function has a variable with same name.

Q. 3. What are the differences between parameters and arguments?

Q. 4. What are default arguments?

Q. 5. What are keyword arguments?

Q. 6. What are the advantages of keyword arguments?

Q. 7. Write a generator function generatesq () that displays the square roots of numbers from 100 to n where n is passed as an argument.

Q. 8. What are the advantages of dividing a program into modules.

Q. 9. Differentiate between Built-in functions and user defined functions.

Q. 10. Differentiate between Built-in functions and functions defined in modules.

#### **Long Answer Type Questions - (3 mark for correct answer)**

Q. 1. List a type of arguments and explain any 2 type of arguments.

Q. 2. Write a method in Python to find and display the prime number between 2 to N. Pass N as argument to the method.

Q. 3. Write a program that uses a function which takes two string arguments and returns the string comparison result of the two passed strings.

Q. 4. Write definition of a function

1. OddSum(Numbers) to add Odd values in the list Numbers.

2. EvenSum(Numbers) to add Even values in the list Numbers.

Q.5. Define a function overlapping () that takes two lists and returns true if they have at least one member in common, False otherwise.

Q.6. Write a program for nth multiple of Fibonacci Series. Also show proper documentation.

Q. 7. Write a Python program to reverse a string.

Q.8. A function checkMain() defined in module Allchecks.py is being used in two different programs

In program 1 as

Allchecks.checkMain(3,'A')

and in program 2 as

checkMain(4,'Z').

Why are the functions call statements different in each program?

Q. 9. Write a python program to find simple interest using a user defined function with parameters and with return value.

Q. 10. Explain any three string functions with example?

### **Case Base Questions - (5 mark for correct answer)**

Q. 1. 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 sentence. 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.

Q. 2. 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 drivers' 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 user enters anything other than RED, YELLOW and GREEN. Function light() is called and the following is displayed depending upon return value from light():

(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.

(c) Display "BETTER LATE THAN NEVER" after the function trafficLight() is executed.

## **Summary**

- A module is a separately saved unit whose functionality can be reused at will.
- A function is a named block of statements that can be invoked by its name.
- Python can have three types of functions:
  - Built-in functions,
  - Functions in modules, and
  - User-defined functions.
- A Python module can contain objects like docstrings, variables, constants, classes, objects, statements, functions
- A Python module has the .py extension.
- A Python module can be imported in a program using import statement.
- There are two forms of Importing Python module statements:
  - (i) import <modulename>
  - (ii) from <module> import <object>
- The built-in functions of Python are always available; one need not import any module for them.
- The math module of Python provides math functionality.
- Functions make program handling easier as only a small part of the program is dealt with at a time, thereby avoiding ambiguity.
- The values being passed through a function-call statement are called arguments (or actual parameters or actual arguments).
- The values received in the function definition/header are called parameters (or formal parameters or formal arguments).
- Keyword arguments are the named arguments with assigned values being passed in the function-call statement. A function may or may not return a value.
- A void function internally returns legal empty value None. The program part(s) in which a particular piece of code or a data value (e.g., variable) can be accessed is known as Variable Scope.
- In Python, broadly, scopes can either be global scope or local scope.
- A local variable having the same name as that of a global variable hides the global variable in its function.
- A file that contains a collection of related functions grouped together and other definitions is called module.
- A search path is the list of directories that the interpreter searches before importing a module.
- A library is just a module that contains some useful definitions.
- The random() function generates a floating point random value from 0 to <1.
- A function is said to be recursive if it calls itself.
- There are two cases in each recursive function—the recursive case and the base case. An infinite recursion is when a recursive function calls itself endlessly.
- If there is no base case or if the base case is never executed, infinite recursion occurs.
- Iteration uses the same memory space for each pass contrary to recursion where fresh memory is allocated for each successive call.
- Recursive functions are relatively slower than their iterative counterparts. Some commonly used recursive algorithms are factorial, gcd, fibonacci series printing, binary search, etc.
- String can be passed to a function as argument but it is used as pass by value.
- Tuple value cannot be modified in a function.
  - In Python, everything is an object, so the dictionary can be passed as an argument to a function like other variables are passed.

### **Short Answer Type Questions – (1 mark for correct answer)**

Q.1 Find and write the output of the following python code:

```
a=10
def call():
 global a
 a=15
call()
print(a)
```

Ans. 15

Q.2. What do you mean by modularity?

Ans. The act of partitioning a program into individual components (modules) is called modularity.

Q.3. What is a function call?

Ans. To use a function that has been defined earlier, a function call statement is written in Python.

Its syntax is <function name> (<value to be passed to argument>)

e.g. print ()

Q.4. Name the three categories of functions.

- Ans.
1. Built-in-Functions
  2. Functions defined in modules
  3. User defined Functions.

Q.5. What is the role of an argument of a function?

Ans. An argument passes the value to the function to work upon.

Q.6. What are docstrings?

Ans. The docstrings are triple quoted strings in a Python module program which are displayed as document when help (<module or program name>) command is issued. (1 mark for correct answer) 1

Q.7. What are docstring conventions?

Ans. General docstring conventions are:

- (1) First Letter of the first line is a capital letter.
- (2) Second line as blank line.
- (3) Rest of the details begin from third line.

Q.8. What is the use of following functions?

- (a) ceil
- (b) sqrt
- (c) exp
- (d) fabs

Ans (a) ceil() function returns smallest integer not less than the argument.

(b) sqrt() function returns square-root of the argument.

(c) exp() function returns natural logarithm e raised to the argument power.

(d) fabs() function returns the absolute value of the supplied argument.

Q.9. Name the constant available in math module.

Ans. math.pi (where pi(#)=3.1415....)

math.e (where e = 2.718281.....)

Q. 10. Write two ways in which you are able to use constant pi in your programs.

Ans. 1st method

```
import math m
```

```
print (math.pi)
```

2nd method

```
from math import pi
print (pi)
```

Q. 11. What is dot notation?

Ans. When a module is imported, we can use its functions or constants by specifying the name of the module and of the function, separated by a dot.

e.g. <module name>. <function name>()

This is called dot notation.

Q.12. What is a function?

Ans. A function is a block of statements that is given a name. This name can be used to invoke the function.

Q.13. What is an argument?

Ans. An argument is data passed to a function through function call statement.

Q. 14. What is the general syntax for defining a function in Python?

Ans. def<function name> ([parameters]):

```
["""<functions docstrings>"""]
<statement>
[<statement>]
```

Q. 15. What is function header?

Ans. The first line of function definition that specifies It begins name of the functions and parameters. with keyword def and ends with a colon().

Syntax: def <function\_name> ([parameters]);

Q.16. What are parameters?

Ans. Parameters are variables listed within parentheses of a function header.

Q.17. The convention for indentation within a block is four spaces. Is it true?

Ans. Yes.

Q. 18. Find the error in the following codes.

```
def minus(total_decrement)
 output= total_decrement
```

Ans. Colon (:) is missing in function header. It should be  
def minus(total\_decrement):

Q. 19. What is \_\_main\_\_?

Ans. The segment with top level statements is named as \_\_main\_\_ by Python.

Q. 20. What is \_\_name\_\_?

Ans. \_\_name\_\_ is a built in variable that states the name of the top level statements i.e. \_\_main\_\_.

Q. 21. What is the difference between arguments and parameters?

Ans. Value that are being passed are called arguments and values that are received are called parameters.

Q. 22. Name two types of function in Python.

Ans. Non-void and void functions.

(½ mark for each function type) 1

Q. 23. Trace the flow of execution for following program.

```
1 def power (b,p):
2 r = b**p
3 return r
4
5 def calcsquare(a):
6 a= power(a,2)
7 return a
8
9 n=5
10 result = calcsquare(n)
11 print(result)
```

Ans. 1→5→9→10→6→2→3→7→11

Q. 24. What will be the output of the following code?

```
def addEm(x,y,z):
 print(x+y+z)

def prod(x,y,z):
 return x*y*z

a= addEm(6,16,26)
b= prod(2,3,6)
print(a,b)
```

Ans. 48

None 36.

Q.25. What is the use of return statement?

Ans. The return statement terminates the execution of a function and returns control to the calling function.

### **Short Answer Type Questions - (2 mark for correct answer)**

Q. 1. Find and write the output of the following Python code:

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

Ans. SCHOOLbbbbCOM

Q. 2. What do you understand by local and global scope of variables? How can you access a global variable inside the function, if function has a variable with same name.

Ans. A global variable is a variable that is accessible globally. A local variable is one that is only accessible to the current scope, such as temporary variables used in a single function definition. A variable declared outside of the function or in global scope is known as global variable. This means, global variable can be accessed inside or outside of the function whereas local variable can be used only inside the function. We can access by declaring variable as global A.

Q. 3. What are the differences between parameters and arguments?

Ans.

| S.No. | Parameters                                 | Arguments                                                                    |
|-------|--------------------------------------------|------------------------------------------------------------------------------|
| 1     | Values provided in function header         | Values provided in function call.                                            |
| 2     | (eg) def area (r):<br>→ r is the parameter | (eg) def main()<br>radius = 5.0<br>area (radius)<br>→ radius is the argument |

Q. 4. What are default arguments?

Ans. Python allows function arguments to have default values; if the function is called without the argument, the argument gets its default value.

Q. 5. What are keyword arguments?

Ans. If there is a function with many parameters and we want to specify only some of them in function call, then value for such parameters can be provided by using their names instead of the positions.

These are called keyword arguments.

```
(e.g.) def simpleinterest(p, n=2, r=0.6):
 def simpleinterest(p, r=0.2, n=3):
```

Q.6. What are the advantages of keyword arguments?

Ans. It is easier to use since we need not to remember the order of the arguments. We can specify the values for only those parameters which we want, and others will have default values.

Q.7. Write a generator function generatesq () that displays the square roots of numbers from 100 to n where n is passed as an argument.

```
Ans. import math
 def generates(n):
 for i in range(100,n):
 print(math.sqrt(i))
```

Q. 8. What are the advantages of dividing a program into modules.

Ans. The advantages of dividing a program into modules :

- It reduces complexity of the program.
- It creates well-defined boundaries within the program.
- It increases the reusability of the module.

Q. 9. Differentiate between Built-in functions and user defined functions.

Ans. Built in functions are predefined functions that are already defined in Python and can be used anytime.

e.g. len(), type(), int(), etc.

User defined functions are defined by the programmer.

Q. 10. Differentiate between Built-in functions and functions defined in modules.

Ans. Built-in functions are predefined functions that are already defined in Python and can be used anytime e.g. len(), type(), int() etc. Functions int defined in

modules are predefined in modules and can be used only when the corresponding module is imported e.g. to use predefined function sqrt() the math module needs to be imported as import math.

### **Long Answer Type Questions - (3 mark for correct answer)**

Q. 1. List a type of arguments and explain any 2 type of arguments.

Ans. Type of arguments:

- Positional arguments
- Default arguments
- Keyword arguments
- Variable length arguments

#### **Keyword arguments :**

If there is a function with many parameters and we want to specify only some of them in function call, then value for such parameters can be provided by using their names instead of the positions.

These are called keyword arguments.

e.g.      def simpleinterest(p, n=2, r=0.6):  
              def simpleinterest(p, r=0.2, n=3):

#### **Default arguments**

Python allows function arguments to have default values; if the function is called without the argument, the argument gets its default value.

e.g.      def add(a,b=0)  
              def mul(a=1,b=1)

(Note:Student may explain any 2 type of argument)

Q. 2. Write a method in Python to find and display the prime number between 2 to N. Pass N as argument to the method.

Ans. def prime(N):

```
 for a in range (2, N):
 prime = 1
 for i in range (2, a):
 if a%i ==0:
 prime = 0
 if prime == 1:
 print (a)
```

**OR**

```
def prime(N):
 for a in range (2, N):
 for i in range (2, a):
 if a%i == 0:
 break
 else:
 print (a)
 break
```

Q. 3. Write a program that uses a function which take two string arguments and returns the string comparison result of the two passed strings.

Ans. def stringComp (str1, str2):  
      if str1.length()!=str2.length:  
          return(False)

```

else:
 for i in range (str1.length()):
 if str1[i]== str2[i]:
 return(False)
 else:
 return (True)

fstring=input ("Enter First string:")
sstream=input ("Enter Second string:")
if stringComp (fstring, sstring):
 print ("Strings are same.")
else:
 print ("Strings are different")

```

Q. 4. Write definition of a function

1. OddSum(Numbers) to add Odd values in the list Numbers.
2. EvenSum(Numbers) to add Even values in the list Numbers.

Ans. def OddSum(Numbers):

```

Sum=0
for i in range(len(Numbers)):
 if(Numbers[i]%2!=0):
 Sum+=Numbers[i]
print(Sum)

```

```

def EvenSum(Numbers):
 Sum=0
 for i in range(len(Numbers)):
 if(Numbers[i]%2==0):
 Sum+=Numbers[i]
 print(Sum)

```

Q.5. Define a function overlapping () that takes two lists and returns true if they have at least one member in common, False otherwise.

Ans. def overlapping (list1, list2):

```

l1= len(list1)
l2= len(list2)
flag=False
for i in range (11):
 for j in range (12):
 if list1[i]==list2[j]:
 flag=True
return flag

```

Q. 6. Write a program for nth multiple of Fibonacci Series. Also show proper documentation.

Ans. # Python Program to find position of nth multiple

# of a number k in Fibonacci Series

def findPosition(k, n):

```

f1 = 0
f2=1
i=2
while i!=0:
 f3=f1+f2

```

```

f1 = f2
f2=f3
if f2%k == 0:
 return n*i
i+=1
return
Multiple no.
n = 5
#Number of whose multiple we are finding
k = 4

print("Position of nth multiple of k in")
print("Fibonacci series is", findPosition(k,n))

```

Q. 7. Write a Python program to reverse a string.

Ans. def string\_reverse(str1):  
 rstr1 = ""  
 index = len(str1)  
  
 while index > 0:  
 rstr1 += str1[index-1]  
 index = index-1  
 return rstr1  
print(string\_reverse("1234abcd"))

Q. 8. A function checkMain() defined in module Allchecks.py is being used in two different programs

In program 1 as

Allchecks.checkMain(3,'A')

and in program 2 as

checkMain(4,'Z').

Why are the functions call statements different in each program?

Ans. In program 1, the complete module Allchecks.py must have been imported as  
**import Allchecks**

So a SpaceName is created and we need to specify the module name.

In program 2, only the function checkMain() must have been imported as

### **From Allchecks import checkMain()**

So the checkMain() function is imported into the namespace of the program2 and hence can be used independently.

Q. 9. Write a python program to find simple interest using a user defined function with parameters and with return value.

Ans. #Python program to calculate simple interest using function

```

def simpleInterest(P, N, R):
 SI = (P * N * R)/100
 return SI

P = float(input("Enter the principal amount : "))
N = float(input("Enter the number of years : "))
R = float(input("Enter the rate of interest : "))
#calculate simple interest by using this formula

```

```

SI = simpleInterest(P, N, R)
#print
print("Simple interest : ",SI)

```

Q. 10. Explain any three string functions with example?

Ans. 1) **isupper()**: The isupper() method returns True if all the characters are in upper case, otherwise False.

```
txt = "THIS IS NOW!"
```

```
x = txt.isupper()
```

```
print(x)
```

2) **upper()** : The upper() method returns a string where all characters are in upper case.

```
txt = "Hello my friends"
```

```
x = txt.upper()
```

```
print(x)
```

3) **isdigit()** : The isdigit() method returns True if all the characters are digits, otherwise False.

```
txt = "50800"
```

```
x = txt.isdigit()
```

```
print(x)
```

**(Note: Student may write any three string functions)**

### **Case Base Questions - (5 mark for correct answer)**

Q. 1. 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 sentence. 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.

Ans. #A user-defined function that accepts a string

# and calculates the number of upper case letters and lower case letters

```
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")
```

Q. 2. 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 drivers' 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 user enters anything other than RED, YELLOW and GREEN. Function light() is called and the following is displayed depending upon return value from light():

- (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.

(c) Display "BETTER LATE THAN NEVER" after the function trafficLight() is executed.

Ans. # 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)# function call to light()
 if(value==0):
 print ("STOP, Life is more important than speed")
 elif (value==1):
 print ("PLEASE GO SLOW.")
 else:
 print ("You may go now.")
def light (colour):
 if (colour == "RED"):
 return (0)
 elif (colour == "YELLOW"):
 return (1)
 else:
 return (2) #function end a here
trafficlight ()
print ("BETTER LATE THAN NEVER")
```

### **Files – Text Files**

A text file stores information in the form of a stream of ASCII or Unicode characters. In text files, each line of text is terminated, (delimited) with a special character known as EOL (End of Line) character. In Python, by default, this EOL character is the newline character ('\n') or carriage-return, newline combination ('\r\n').  
The text files can be of following types:

- Regular Text files:** These are the text files which store the text in the same form as typed. Here the newline character ends a line and the text translations take place. These files have a file extension as .txt.
- Delimited Text files:** In these text files, a specific character is stored to separate the values, i.e., after each value, e.g., a tab or a comma after every value.
  - When the comma is used to separate the values stored, these are called CSV files (Comma Separated Values files). These files take the extension as .csv.

## **Files – Binary Files**

A binary file stores the information in the form of a stream of bytes. A binary file contains information in the same format in which the information is held in memory, i.e., the file content that is returned to you is raw (with no translation or no specific encoding). As a result, binary files are faster and easier for a program to read and write than are text files. As long as the file doesn't need to be read by people or need to be ported to a different type of system, binary files are the best way to store program information.

**File Object:** A file object is a reference to a file on disk. It opens the file and makes it available for a number of different tasks.

### **Absolute Vs Relative path:**

An absolute path is defined as specifying the location of a file or directory from the root directory. In other words, we can say that an absolute path is a complete path from start of actual file system from / directory.

Relative path is defined as the path related to the present working directly.

**File Modes:** A file mode govern the type of operations (read/write/append) possible in the opened file.

| Sr.No. | Modes & Description                                                                                                                                                                                          |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1      | <b>r:</b> Opens a file for reading only. The file pointer is placed at the beginning of the file. This is the default mode.                                                                                  |
| 2      | <b>r+:</b> Opens a file for both reading and writing. The file pointer placed at the beginning of the file.                                                                                                  |
| 3      | <b>w:</b> Opens a file for writing only. Overwrites the file if the file exists. If the file does not exist, creates a new file for writing.                                                                 |
| 4      | <b>w+:</b> Opens a file for both writing and reading. Overwrites the existing file if the file exists. If the file does not exist, creates a new file for reading and writing.                               |
| 5      | <b>a:</b> Opens a file for appending. The file pointer is at the end of the file if the file exists. That is, the file is in the append mode. If the file does not exist, it creates a new file for writing. |

|   |                                                                                                                                                                                                                                      |
|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 6 | <b>a+:</b> Opens a file for both appending and reading. The file pointer is at the end of the file if the file exists. The file opens in the append mode. If the file does not exist, it creates a new file for reading and writing. |
|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

### **File Read Methods:**

| Sr.No. | Methods & Description                                                                                                                                                                                                      |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1      | <b>Filehandle.read([n]):</b> reads and return n bytes, if n is not specified it reads entire file.                                                                                                                         |
| 2      | <b>Filehandle.readline([n]):</b> reads a line of input. If n is specified reads at most n bytes. Read bytes in the form of string ending with end of line character or blank string if no more bytes are left for reading. |
| 3      | <b>Filehandle.readlines():</b> reads all the lines and return them in a list.                                                                                                                                              |

### **File Write Methods:**

| Sr.No. | Methods & Description                                                                                   |
|--------|---------------------------------------------------------------------------------------------------------|
| 1      | <b>Filehandle.write(str):</b> Write string str to file referenced by filehandle.                        |
| 2      | <b>Filehandle.writelines(L):</b> Write all strings in list L as lines to file referenced by filehandle. |

### **Opening text file:**

open() function: The open() function is used to open a file in the following manner:

```
<file-objectname> = open(<filename>)
<file-objectname> = open(<filename>,<mode>)
```

By default the file open mode is read mode.

with statement: with statement will automatically close the file after the nested block of code. It is guaranteed to close the file no matter how the nested block exits. Even if an exception occurs with statement will handle it and close the file. It is used in the following manner:

```
with open(<filename>,<filemode>) as <filehandle>:
 <file manipulation statements>
```

**Closing a text file:** A close() function breaks the link of file-object and the file on the disk. No tasks can be performed after the file is closed. It is used in the following way:

```
<file-handle>.close()
```

**The tell() Function:** The tell() function returns the current position of the file pointer in a file. It is used as per the following syntax:

<file-object>.tell()

**The seek() Function:** the seek() function changes the position of the file pointer by placing the file pointer at the specified position in the opened file. It is used as per the following syntax:

<file-object>.seek(offset[, mode])

offset: is a number specifying number of bytes.

mode: 0 – for beginning of the file – by default 0

1 – for current position of file pointer

2 – for end of the file

### **Questions – 1 Mark - MCQ**

**Q1.** To open a file c:\\ss.txt for appending data we use

- a. file = open('c:\\\\ss.txt','a')
- b. file = open(r'c:\\ss.txt','a')
- c. file = open('c:\\\\ss.txt','w')
- d. both a and b

**Q2.** To read the next line of the file from the file object infi, we use

- a. infi.read(all)
- b. infi.read()
- c. infi.readline()
- d. infi.readlines()

**Q3.** Which function is used to ensure that the data is written in the file immediately?

- a. <filehandle>.write()
- b. <filehandle>.writelines()
- c. flush()
- d. <filehandle>.close()

**Q4.** What is the datatype of the value returned by readlines() function?

- a. Integer
- b. String
- c. List of strings
- d. None of the above

**Q5.** What is the position of the cursor when the file is opened in append mode?

- a. Start of file
- b. End of file
- c. At the 11<sup>th</sup> byte
- d. Unknown

**Q6.** How to open a file such that close function is not needed to be called in order to close the connection between file and python program?

- a. Using open() method
- b. Using read() method
- c. Using with keyword
- d. Using open with method

**Q7.** What is the full form of CSV?

- a. Common segregated values
- b. Comma separated values
- c. Common separated values
- d. None of the above

**Q8.** If a file is opened for writing

- a. File must exist before opening
- b. File will be created if does not exist
- c. None of the above
- d. Both a and b

**Q9.** If the cursor in the file is at the end of the file, then what will the read() function return?

- a. None
- b. False
- c. Exception
- d. Empty string

**Q10.** Which is the value of mode parameter to set the offset of the cursor from the end of the file?

- a. 0
- b. 1
- c. 2
- d. None of the above

|            |     |     |     |     |     |     |     |     |      |
|------------|-----|-----|-----|-----|-----|-----|-----|-----|------|
| Answer Key |     |     |     |     |     |     |     |     |      |
| 1.d        | 2.c | 3.c | 4.c | 5.b | 6.c | 7.b | 8.a | 9.d | 10.c |

**Q1.** What is the difference between opening mode 'a' and 'w'?

**Q2.** What is the purpose of flush() in file handling operations?

**Q3.** What is the advantage of opening file using with keyword?

**Q4.** Consider the following file 'corona.txt'

O corona O corona  
Jaldi se tum Go na  
Social Distancing ka palan karona  
Sabse 1 meter ki duri rakho na

Write the output of the following statement:

```
f = open('corona.txt')
str1 = _____ # to read first line
str2 = _____ # to read next line of file
str3 = _____ # to read remaining lines of the file
```

**Q5.** Consider the following file 'corona.txt'

O corona O corona  
Jaldi se tum Go na  
Social Distancing ka palan karona  
Sabse 1 meter ki duri rakho na

Complete the missing statement using for loop to print all the lines of code:

```
f = open('corona.txt')
for _____:
 print(____)
```

**Q6.** Given a file 'data.txt' write a function atoedisp() to display the file after replacing 'a' with 'e'.

**Q7.** What is the advantage of saving a file in text form and binary form?

**Q8.** What is the output of following code?

```
fh = open('main.txt','r')
size = len(fh.read())
print(fh.read())
```

**Q9.** Write the output of the following program

```
f = open('data.txt', 'w+')
f.write('0123456789abcdef')
f.write('xyz1234')
f.seek(0)
print(f.read())
f.close()
```

**Q10.** If a file 'f.txt' contains data 'Good' and then what is the content of the file f.txt then following program in run?

```
f = open('f.txt', 'w')
f.write('Bye')
f.close()
```

| S.No | Answer Key                                                                                                                                                                                                                                                                                    |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | 'w' is used to write in file from the beginning. If file already exists then it will overwrite the previous content.<br>'a' is also used to write in the file. If file already exists it will write after the previous content.                                                               |
| 2    | The flush function forces the writing of data on disk still pending in output buffer. Flush allows the user to send the content in file before closing the file.                                                                                                                              |
| 3    | With keyword reduces the overheads involved in the closing of the file after file operation or handling the file closing after exception have occurred. When file is opened using 'with' it will manage the closing of file at the end of the operations or after an exception automatically. |
| 4    | f.readline()<br>f.readline()<br>f.readlines()                                                                                                                                                                                                                                                 |
| 5    | for line in f:<br>print(line)                                                                                                                                                                                                                                                                 |
| 6    | def atoedisp():<br>f = open('data.txt')<br>r = f.read()<br>for ch in r:<br>if ch == 'a':<br>print('e', end = '')<br>else:<br>print(ch, end = '')                                                                                                                                              |

|    |                                                                                                                                                                                                                                    |
|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7  | Text files are stored in the human readable format. These have some internal conversions like newlines etc.<br>Binary file store data in pure binary form and hence can be used on any machine/hardware with appropriate software. |
| 8  | No output (Empty string).<br>fh.read() in the second line puts the cursor at the end of the file. That is why the read function in the third line will return an empty string.                                                     |
| 9  | 0123456789abcdefxyz1234                                                                                                                                                                                                            |
| 10 | Bye<br>Existing data will be overwritten.                                                                                                                                                                                          |

### **Questions – 3 Marks – Long Answer**

- Q1.** Differentiate between 'r' and 'w' file open modes and 'w' and 'a' file open mode.
- Q2.** Write a function stats() that accepts a filename and reports the files longest line.
- Q3.** A text file contains alphanumeric text (say an.txt). Write a program that reads this text file and prints only the numbers or digits from the file.
- Q4.** Write a program to read a text file and display the count of lowercase and uppercase letters in the file.
- Q5.** Write a method in python to read lines from a text file INDIA. TXT, to find and display the Occurrence of the word "India".
- Q6.** Write a program that copies a text file "source.txt" onto "target.txt" barring the starting with a "@" sign.
- Q7.** Take a sample text file and find the most commonly occurring word. Also, list the frequencies of words in the text file.
- Q8.** Write a program to read a text file line by line and display each word separated by a #.
- Q9.** Write a function remove\_lowercase() that accepts two filenames, and copies all the lines that do not start with a lowercase letter from the first file into the second.
- Q10.** Write a program to display all the records in a file along with line/record number.

| S.No | Answer Key                                                                                                                                                                                                                |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.   | Refer the notes above.                                                                                                                                                                                                    |
| 2.   | <pre>def stats(filename):     longest = ""     for line in file(filename):         if len(line) &gt; len(longest):             longest = line     print("Longest line's length =", len(longest))     print(longest)</pre> |
| 3.   | <pre>F open("an. txt", "r") for line in F:     words line.split()     for i in words:</pre>                                                                                                                               |

|    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <pre> for letter in i:     if( letter.isdigit () ) :         print(letter) </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 4. | <pre> myfile = open( "Answer.txt", "r") ch = " "                      #initially stored a space ( a non-None value) lcount = 0                      #variable to store count of lowercase letters ucount 0 while ch:                      #while ch stores a Non-None value     ch = myfile.read(1) #one character read from file     if ch.isupper() == True:         ucount ucount+1     else:         lcount = lcount + 1 print("Uppercase letters in the file: ", ucount) print ("Lowercase letters in the file: ", lcount) #close the file myfile.close() </pre> |
| 5. | <pre> def display1():     count = 0     file = open( 'INDIA. TXT', 'r')     for LINE in file:         Words LINE. split()         for W in Words             if W= "India":                 count count +1     print (count)     file.close() </pre>                                                                                                                                                                                                                                                                                                                  |
| 6. | <pre> def filter (oldfile, newfile):     fin = open(oldfile, "r")     fout open(newfile, "w")     while True:         text fin.readline()         if len(text) != 0:             if text[e] != "@":                 fout.write(text)     fin.close()     fout.close() filter("source.txt", "target.txt") </pre>                                                                                                                                                                                                                                                       |
| 7. | <pre> with open ("nanthem.txt", "r") as fh:     contents = fh.read() Wordlist = contents.split() Wordfreq= [] High = 0 Word = '' existing = [] for w in wordlist:     wcount = wordlist.count(w)     if w not in existing:         wordfreq.append( [w,wcount])         existing.append(w)         if wcount&gt; high:             high = wcount             Word = w </pre>                                                                                                                                                                                          |

|     |                                                                                                                                                                                                                                                                                                |
|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|     | <pre> print("The word ' "+ word+" ' occurs maximum number of times, ", high, "times. ") print("\n other words have these frequencies: ") print (wordfreq) </pre>                                                                                                                               |
| 8.  | <pre> myfile = open( "Answer.txt", "r") line = " " while line:     line = myfile.readline()      # one line read from file     # printing the line word by word using split()     for word in line.split():         print (word, end = '#*')     print() #close the file myfile.close() </pre> |
| 9.  | <pre> def remove_lowercase (infile, outfile):     output file (outfile, "w")     for line in file(infile):         if not line[e] in "abcdefghijklmnopqrstuvwxyz" :             output.write(line) output.close () </pre>                                                                      |
| 10. | <pre> fh = open("Result.det", "r") count = 0 rec = "" while True:     rec fh.readline()     if rec != "":         count count +1         print(count, rec) fh.close() </pre>                                                                                                                   |

### **Questions – 5 Marks – Case Based Questions**

**Q1.** Dhanush's teacher has given him the following text file (HigherEd. txt).

Higher education improves an individual's quality of life.

Studies show that, compared to high school graduates, college graduates have longer life spans, better access to health care, better dietary and health practices, greater economic stability and security.

We must ensure that our whole population receives an education that will allow full and continuing participation in this dynamic period of economic history.

And the following incomplete code:

```

def fileFunction1(_____) #Fil1_line5
 fin = _____ #Fi11_line6
 print(_____(N1)) #Fi11_line7
 _____ #Fil1_line8
 print(_____(N2)) #Fil1_line9
#__main__
N1 = _____ #Fill_Line1
N2 = _____ #fill_Line2
Filename = _____ #Fill_Line3
fileFunction1 (Filename, N1, N2) #Line4

```

Help Dhanush to complete his work as per the following instructions:

- Add code to blank lines Fill\_Linel and Fill\_Line2 so that two integer numbers are read into variables N1 and N2.

Also, complete Fill\_Line3 so that the function call below it makes sense, which is sending 3 arguments : filename, and two numbers N1 and N2.

- As per the function call, complete the function header in Fill\_Line5.
- Complete Fill\_Line6 so that the text file is opened for reading in the file object fin.
- Complete Fill\_Line7 and Fill\_Line9 so that the passed number of characters are read from the file and printed.
- Complete Fill\_Line8 so that the characters till the end of the line from the current position of the file pointer are read but not printed.

**Q2.** SHRUTHI has to complete her file-based assignment by tonight. She has been given the following text file (Education.txt):

```
Higher education improves quality of life.
College graduates have longer life spans.
Education is birthright.
```

Shruthi has also received the following incomplete code.

```
def fileFunction1 (_____,_____) #Fill_Line5
 fi = _____(fname,_____) #Fill_Line6
 fi._____ #Fill_Line7
 fi._____ #Fill_Line8
 fi.close()
 print("Done")

def fileFunction2(fname, N1, N2):
 fi = open(fname)
 print(fi.read(N1))
 fi.readline()
 print(fi.read(N2))
 a = fi.readlines ()
 print(a)
N1 = 16 #Line1
N2 = 22 #Line2
String = "India strengthening" #Line3
fileFunction1(_____) #Fill_Line4
fileFunction2 ('Education.txt', N1, N2)
```

Help her to complete her assignment as per the following instructions.

- Complete Fill\_Line4 so that the function call to FileFunction1() passes two arguments: First as the filename and the second as the string given in Line 3.
- Complete Fill\_Line5 so that the function header is as per its function call.
- Complete Fill\_Line6 so that the file is opened in a mode that will allow it to write the string at the end of the file without deleting anything from the file.
- Complete Fill\_Line7 and Fill\_Line8 so that the passed string is written on to the file, followed by a newline character.
- What will be the output produced by the complete code?

|       |            |
|-------|------------|
| S. No | Answer Key |
|-------|------------|

|   |                                                                                                                                                                                                                                                                                                        |
|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | <pre> a. N1 = int(input ("Enter number1: ")) N2 = int(input ("Enter number 2:")) fileName= "HigherEd. txt" fileFunction1 (fileName, N1, N2) b. def fileFunction1 (fname, N1, N2): c. fin open(fname, "") d. print(fin.read(N1)) print(fin.read(N2)) e. fin.readline() </pre>                           |
| 2 | <pre> a. fileFunction1(" Education.txt', string) b. def fileFunction1(fname, string): c. fi open(fname, "a") d. fi.write(string) fi.write('\n') e. Done </pre> <p>Higher education<br/>college graduates have<br/>['longer life spans.\n', 'Education is birth right.\n', 'India strengthening\n']</p> |

Binary files store data in the binary format (0's and 1's) which is understandable by the machine. So when we open the binary file in our machine, it decodes the data and displays in a human-readable format.

**Binary File Modes:** File mode governs the type of operations read/write/append possible in the opened file. It refers to how the file will be used once its opened.

| File Mode | Description                                                                                                              |
|-----------|--------------------------------------------------------------------------------------------------------------------------|
| rb        | Read Only: Opens existing file for read operation                                                                        |
| wb        | Write Only: Opens file for write operation. If file does not exist, file is created. If file exists, it overwrites data. |
| ab        | Append: Opens file in write mode. If file exist, data will be appended at the end.                                       |
| rb+       | Read and Write: File should exist, Both read and write operations can be performed.                                      |
| wb+       | Write and Read: File created if not exist, If file exist, file is truncated.                                             |
| ab+       | Write and Read: File created if does not exist, If file exist data is truncated.                                         |

### Write data to a Binary File:

Pickle is a special python package that is used to generate data in binary format. Pickle comes with few methods like dump( ) to write data in binary format.

Example:

```

import pickle
list =[] # empty list
while True:
 roll = input("Enter student Roll No:")
 sname = input("Enter student Name :")
 student = {"roll":roll,"name":sname} # create a dictionary
 list.append(student) # add the dictionary as an element

```

```

in the list
 choice= input("Want to add more record(y/n) :")
 if(choice=='n'):
 break
file = open("student.dat","wb") # open file in binary and write
mode pickle.dump(list, file)
file.close()

```

**OUTPUT:**

```

Enter student Roll No:1201
Enter student Name :Anil
Want to add more record(y/n) :y
Enter student Roll No:1202
Enter student Name :Sunil
Want to add more record(y/n) :n

```

### **Read data from a Binary File:**

To read the data from a binary file, we have to use load( ) function of pickle module.

**Example:**

```

import pickle
file = open("student.dat", "rb")
list = pickle.load(file)
print(list)
file.close()
OUTPUT:
[{'roll': '1201', 'name': 'Anil'}, {'roll': '1202', 'name': 'Sunil'}]

```

### **Update a record in Binary File:**

Locate the record to be updated by searching for it.

Make changes in the loaded record in memory

Write back onto the file at the exact location of record.

```

import pickle
roll = input('Enter roll number whose name you want to update in binary file :')
file = open("student.dat", "rb+")
list = pickle.load(file)
found = 0
lst = []
for x in list:
 if roll in x['roll']:
 found = 1
 x['name'] = input('Enter new name: ')
 lst.append(x)
if found == 1:
 file.seek(0)
 pickle.dump(lst, file)
 print("Record Updated")
else: print('roll number does not exist')
file.close()
OUTPUT:
Enter roll number whose name you want to update in binary file :1202
Enter new name: Harish

```

Record Updated

### **Delete a record from binary file:**

```
import pickle
roll = input('Enter roll number whose record you want to delete:')
file = open("student.dat", "rb+")
list = pickle.load(file)
found = 0
lst = []
for x in list:
 if roll not in x['roll']:
 lst.append(x)
 else:
 found = 1
if found == 1:
 file.seek(0)
 pickle.dump(lst, file)
 print("Record Deleted ")
else:
 print('Roll Number does not exist')
file.close()
```

#### **OUTPUT:**

Enter roll number whose record you want to delete:1201  
Record Deleted

### **Search a record in a binary file:**

```
import pickle
roll = input('Enter roll number that you want to search in binary file :')
file = open("student.dat", "rb")
list = pickle.load(file)
file.close()
for x in list:
 if roll in x['roll']:
 print("Name of student is:", x['name'])
 break
 else:
 print("Record not found")
```

#### **OUTPUT:**

Enter roll number that you want to search in binary file :1202  
Name of student is: Harish

### **tell( ) and seek( ) methods:**

tell( ): It returns the current position of cursor in file.

Example:

```
fout=open("story.txt","w")
fout.write("Welcome Python")
print(fout.tell())
fout.close()
```

#### **Output:**

**15**

seek(offset, reference\_point) : Change the cursor position by bytes as specified by

the offset, from the reference point.

Example:

```
fout=open("story.txt","w")
fout.write("Welcome Python")
fout.seek(5)
print(fout.tell())
fout.close()
```

**Output:**

5

**Pickle Module:** Python Pickle is used to serialize and deserialize a python object structure. Any object on python can be pickled so that it can be saved on disk.

**Pickling:** Pickling is the process whereby a Python object hierarchy is converted into a byte stream.

**Unpickling:** A byte stream is converted into object hierarchy.

To use the picking methods in a program, we have to import pickle module using import keyword.

Example:

```
import pickle
```

In this module, we shall discuss to functions of pickle module, which are:

- i. dump( ) : To store/write the object data to the file.
- ii. load( ) : To read the object data from a file and returns the object data.

Syntax:

Write the object to the file:

```
pickle.dump(List_name, file-object)
```

Read the object from a file:

```
pickle.load(file-object)
```

### 1 Mark Questions

1. The process of converting byte stream back to the original structure is known as
  - a. Picklingb. Unpicklingc. Packingd. Zipping
2. Which file mode is used to handle binary file for reading.
  - a. rb
  - b. rw
  - c. r
  - d. w
3. Which of the following is not a correct statement for binary files?
  - a. Easy for carrying data into buffer
  - b. Much faster than other file systems
  - c. Characters translation is not required
  - d. Every line ends with new line character '\n'
4. Which one of the following is correct statement?
  - a. import – pickle
  - b. pickle import
  - c. import pickle
  - d. All the above
5. Which of the following file mode opens a file for append or read a binary file and moves the files pointer at the end of the file if the file already exist otherwise create a new file?
  - a. a
  - b. ab
  - c. ab+
  - d. a+

6. Which of the following file mode opens a file for reading and writing both as well as overwrite the existing file if the file exists otherwise creates a new file?

- a. w              b. wb+              c. wb              d. rwb

7. Mr Sharma is working on a binary file and wants to write data from a list to a binary file.

Consider list object as l1, binary file sharma\_list.dat, and file object as f.

Which of the following can be the correct statement for him?

- a. f = open('sum\_list','wb'); pickle.dump(l1,f)  
b. f = open('sum\_list','rb'); l1=pickle.dump(f)  
c. f = open('sum\_list','wb'); pickle.load(l1,f)  
d. f = open('sum\_list','rb'); l1=pickle.load(f)

8. Every file has its own identity associated with it. Which is known as

- a. icon              b. extension              c. format              d. file type

9. EOL Stands for :

- a. End of Lines      b. End of Line      c. End of List      d. End of Location

10. Which of the following file types allows to store large data files in the computer memory?

- a. Binary Files      b. Text Files      c. CSV Files      d. None of these

## 2 Mark Questions

1. Write a program in python to write and read structure, dictionary to the binary file.

2. BINARY file is unreadable and open and close through a function only so what are the advantages of using binary file

3. Write a statement to open a binary file name sample.dat in read mode and the file sample.dat is placed in a folder ( name school) existing in c drive

4. When do you think text files should be preferred over binary files?

5. Consider a binary file employee.dat containing details such as empno:ename:salary (seperator ':') write a python function to display details of those employees who are earning between 20000 and 30000(both values inclusive)

6. Differentiate between pickle.*load()* and pickle.*dump()* methods with suitable example.

7. A binary file "Book.dat" has structure [BookNo, Book\_Name, Author, Price].Write a user defined function *CreateFile()* to input data for a record and add to Book.dat

8. 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.

9. A binary file "Store.dat" has structure [ItemNo, Item\_Name, Company, Price]. Write a function *CountRec(Company)* in Python which accepts the Company name as parameter and count and return number of Items by the given Company are stored in the binary file "Store.dat".

10. A binary file "Store.dat" has structure [ItemNo, Item\_Name, Company, Price].

Write a function AddRecord(<List>) which accepts a List of the record [ItemNo, Item\_Name, Company, Price] and appends in the binary file "Store.Dat"

### 3 mark Questions

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

i. Write a user defined function CreateFile() to input data for a record and add to "Book.dat".

ii. 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"

2. A binary file "SCHOOL.DAT" has structure [Roll\_Num, Name, Percentage]

i) Write a function Count\_Rec() in Python that would read contents of the file "SCHOOL.DAT" and display the details of those students whose percentage is below 33 .

ii) Write a function Disp\_Rec(alphabet) in Python that would read contents of the file "SCHOOL.DAT" and display the details of those students whose name begin with the alphabet as passed as parameter to the function.

3. A binary file "STOCK.DAT" has structure [ITEMID, ITEMNAME, QUANTITY, PRICE]. Write a user defined function MakeFile( )to input data for a record and add to Book.dat.

4. Write a function GetPrice(ITEMID) in Python which accepts the ITEMID as parameter and return PRICE of the Item stored in Binary file STOCK.DAT.

5. A binary file "EMPLOYEE.DAT" has structure (EMPID, EMPNAME, SALARY).

Write a function CountRec( ) in Python that would read contents of the file "EMPLOYEE.DAT" and display the details of those Employees whose Salary is above 20000.

6. A binary file "EMPLOYEE.DAT" has structure (EMPID, EMPNAME, SALARY). Write a function to display number of employees having Salary more than 20000.

7. A binary file named "EMP.dat" has some records of the structure [EmpNo, EName, Post, Salary], Write a user-defined function named NewEmp() to input the details of a new employee from the user and store it in EMP.dat.

8. Write a user-defined function named SumSalary(Post) that will accept an argument the post of employees & read the contents of EMP.dat and calculate the SUM of salary of all employees of that Post.

9. A binary file named "TEST.dat" has some records of the structure [TestId, Subject, MaxMarks, ScoredMarks] Write a function in Python named DisplayAvgMarks(Sub) that will accept a subject as an argument and read the contents of TEST.dat.
10. Write a python program to search and display the record of the student from a binary file "Student.dat" containing students records (Rollno, Name and Marks). Roll number of the student to be searched will be entered by the user.

### **5 Mark Questions**

1. A binary file "student.dat" has structure [rollno, name, marks]. i. Write a user defined function insertRec() to input data for a student and add to student.dat.  
ii. Write a function searchRollNo( r ) in Python which accepts the student's rollno as parameter and searches the record in the file "student.dat" and shows the details of student i.e. rollno, name and marks (if found) otherwise shows the message as 'No record found'.
2. Write a python program to create binary file dvd.dat and write 10 records in it Dvd id,dvd name,qty,price Display those dvd details whose dvd price more than 25.

### **ANSWERS**

#### **1 Mark Questions Answers**

|          |          |           |          |
|----------|----------|-----------|----------|
| <b>1</b> | <b>A</b> | <b>6</b>  | <b>B</b> |
| <b>2</b> | <b>A</b> | <b>7</b>  | <b>A</b> |
| <b>3</b> | <b>D</b> | <b>8</b>  | <b>B</b> |
| <b>4</b> | <b>C</b> | <b>9</b>  | <b>B</b> |
| <b>5</b> | <b>C</b> | <b>10</b> | <b>A</b> |

#### **2 Mark Question Answers**

```

1.
import pickle
d1={'jan':31,'feb':28,'march':31,'april':30}
f=open('binfile.dat','wb+')
pickle.dump(d1,f)
d2=pickle.load(f)
print(d2)
f.close()

```

2. Binary file are easier and faster than text file. Binary files are also used to store binary data such as images, video files, and audio files.
3. f1=open("c:\school\sample.dat",'r')
4. Text file should be preferred when we have to save data in text format and security of file is not important
5.

```

def Readfile():

```

```
i=open("Employee.dat", "rb+")
x=i.readline()
while(x):
 I= x.split(':')
 if (float(I[2]) >=20000) and (float I[2])<=40000):
 print(x)
 x= i.readline()
```

6.

dump( ) : To store/write the object data to the file.

load( ) : To read the object data from a file and returns the object data.

Syntax:

Write the object to the file:

```
pickle.dump(List_name, file-object)
```

Read the object from a file:

```
pickle.load(file-object)
```

7.

```
import pickle
def createFile():
```

```
fobj=open("Book.dat","ab")
BookNo=int(input("Book Number : "))
Book_name=input("Name : ")
Author = input("Author: ")
Price = int(input("Price : "))
rec=[BookNo,Book_Name,Author,Price]
pickle.dump(rec,fobj)
fobj.close()
```

8.

```
import pickle
def CountRec():
 fobj=open("STUDENT.DAT","rb")
 num = 0
 try:
 while True:
 rec=pickle.load(fobj)
 if rec[2] > 75:
 print(rec[0],rec[1],rec[2],sep="\t")
 num = num + 1
 except:
 fobj.close()
 return num
```

9.

```
import pickle
def CountRec(Cname):
 fobj=open("store.dat","rb")
 num = 0
 try:
 while True:
```

```

rec=pickle.load(fobj)
if Author==rec[2]:
 num = num + 1
except:
 fobj.close()
return num

```

10.

```

def AddRecord(rec):
 print ("Record to be Added is",rec)
 pickle.dump(rec,fobj)
 fobj.close()

```

### **3 Mark questions answers**

1.

```

import pickle
def createFile():
 f=open("Book.dat","ab")
 BookNo=int(input("Book Number : "))
 Book_name=input("Name : ")
 Author = input("Author: ")
 Price = int(input("Price : "))
 rec=[BookNo,Book_Name,Author,Price]
 pickle.dump(rec,f)
 f.close()

```

```

def CountRec(Author):
 f=open("Book.dat","rb")
 num = 0
 try:
 while True:
 rec=pickle.load(f)
 if Author==rec[2]:
 num = num + 1
 except:
 f.close()
 return num

```

2.

```

import pickle
def CountRec():
 fobj=open("SCHOOL.DAT","rb")
 num = 0
 try:
 while True:
 rec=pickle.load(fobj)

```

```

 if rec[2] < 33:
 print(rec[0],rec[1],rec[2],sep="\t")
 num = num + 1
 except:
 fobj.close()
 return num
import pickle
def Disp_Rec(alpha):
 fobj=open("SCHOOL.DAT","rb")
 num = 0
 try:
 while True:
 rec=pickle.load(fobj)
 if rec[1][0] == alpha :
 print(rec[0],rec[1],rec[2],sep="\t")
 except:
 fobj.close()

```

### 3. import pickle

```

def MakeFile():
 fobj=open("STOCK.DAT",'ab')
 itemid=input("Enter ITEM ID")
 itemname=input("Enter name of the Item")
 Q=int(input("Enter the quantity of Item"))
 price=float(input("Enter the price"))
 rec=[itemid,itemname,Q,price]
 pickle.dump(rec,fobj)
fobj.close()

```

### 4. def GetPrice(ITEMID):

```

fobj=open("STOCK.DAT",'rb')
try:
 while True:
 rec=pickle.load(fobj)
 if(rec[0]in str(ITEMID)):
 print("Price of ", rec[0], "item is",rec[3])
 return rec[0]
except:
fobj.close()

```

### 5. import pickle

```

def CountRec():
fobj=open("EMPLOYEE",'rb')
num=0
try:
 while True:
 rec=pickle.load(fobj)
 if(rec[2]>20000):
 print(rec[0],rec[1],rec[2])

```

```
 num=num+1
except:
fobj.close()
return num

6. import pickle
def CountRec():
fobj=open("EMPLOYEE",'rb')
num=0
try:
while True:
 rec=pickle.load(fobj)
 if(rec[2]>20000):
 print(rec[0],rec[1],rec[2])
 num=num+1
except:
fobj.close()
return num
```

7.

```
import pickle

def NewEmp():
 print("Enter the details of an employee:")
 no=int(input("Enter the Empno"))
 name=input("Enter the name")
 post=input("Enter the post")
 sal=float(input("Enter the salary"))

 erec=[no,name,post,sal]
 f=open("EMP.dat","ab")
 pickle.dump(erec,f)
 print("New record saved")
 f.close()
```

8.

```

def SumSalary(Post):
 f=open("EMP.dat", "rb")
 count=0
 sum=0

 try:
 while True:
 rec=pickle.load(f)
 if rec[3]==Post:
 sum+=rec[4]
 except EOFError:
 f.close()

 print("Sum of Salary :", sum)

```

**9.**

```

def DisplayAvgMarks(Sub):
 f=open("ABC.dat", "rb+")
 count=0
 sum=0

 try:
 while True:
 pos=f.tell()
 rec=pickle.load(f)
 print(rec)
 if rec[1]==Sub:
 sum+=rec[3]
 count+=1
 except EOFError:
 f.close()

 print("Average marks scored :", sum/count)

```

**10.**

```

import pickle
f1 = open("E:\Student.dat", "rb")
Stud_rec = pickle.load(f1)
rno = int(input("Enter the roll no to search: "))
flag = 0
for r in Stud_rec:
 if rno == r[0]:
 print (rollno, name, marks)
 flag = 1
if flag == 0:
 print("Record not found...")
f1.close()

```

## 5 Mark Questions answers

1

- i. import pickle

```

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

```

```

ii. def searchRollNo(r):
 f=open("student.dat","rb")
 flag = False
 while True:
 try:
 rec=pickle.load(f)
 if rec[0] == r :
 print(rec['Rollno'])
 print(rec['Name'])
 print(rec['Marks'])
 flag = True
 except EOFError:
 break
 if flag == False:
 print("No record Found")
 f.close()

```

2

```

import pickle
f=open("pl.dat","ab")
ch="Y"
while ch=="Y":
 l=[]
 pi=int(input("enter dvd id "))
 pnm=input("enter dvd name ")
 sp=int(input("enter qty "))
 p=int(input("enter price(in rupees) "))
 l.append(pi)
 l.append(pnm)
 l.append(sp)
 l.append(p)
 pickle.dump(l,f)
 ch=input("do you want to enter more rec(Y/N): ").upper()
 if ch=="Y":

```

```
continue
else:
 break
f.close()
f=open("pl.dat","rb+")
try:
 while True:
 l=pickle.load(f)
 if l[3]>25:
 print(l)
except EOFError:
 pass
f.close()
```

## **Working with CSV Files: CSV (Comma Separated Values)**

A csv file is a type of plain text file that uses specific structuring to arrange tabular data. csv is a common format for data interchange as it is compact, simple and general.

Each line of the file is one line of the table.

csv files have .csv as file extension.

Let us take a **data.csv** file which has the following contents:

Roll No., Name of student, stream, Marks

1, Anil, Arts, 426

2, Sujata, Science, 412

As you can see each row is a new line, and each column is separated with a **comma**.

This is an example of how a CSV file looks like.

To work with csv files, we have to import the **csv module** in our program.

### **Read a CSV file:**

To read data from a CSV file, we have to use reader( ) function. The reader function takes each row of the file and make a list of all columns.

### **CODE:**

```
import csv
with open('C:\\\\data.csv','rt') as f:
 data = csv.reader(f) #reader function to generate a reader object
 for row in data:
 print(row)
```

## **OUTPUT:**

```
['Roll No.', 'Name of student', 'stream', 'Marks']
['1', 'Anil', 'Arts', '426']
['2', 'Sujata', 'Science', '412']
```

## **Write data to a CSV file:**

When we want to write data in a CSV file you have to use writer( ) function.

To iterate the data over the rows (lines), you have to use the writerow( ) function.

## **CODE:**

```
import csv
with open('C:\\\\data.csv', mode='a', newline='') as file:
 writer = csv.writer(file, delimiter=',', quotechar=""") #write new record in file
 writer.writerow(['3', 'Shivani', 'Commerce', '448'])
 writer.writerow(['4', 'Devansh', 'Arts', '404'])
```

## **OUTPUT:**

```
['Roll No.', 'Name of student', 'stream', 'Marks']
['1', 'Anil', 'Arts', '426']
['2', 'Sujata', 'Science', '412']
['3', 'Shivani', 'Commerce', '448']
['4', 'Devansh', 'Arts', '404']
```

When we shall open the file in notepad (Flat file) then the contents of the file will look like this:

```
Roll No.,Name of student,stream,Marks
1,Anil,Arts,426
2,Sujata,Science,412
3,Shivani,Commerce,448
```

## For Practical Explanation

**Program-1**:-Write a program to perform read and write operation with .csv file.

### CODE:

```
import csv
def readcsv():
 with open('C:\\data.csv','rt')as f:
 data = csv.reader(f) #reader function to generate a reader object
 for row in data:
 print(row)

def writecsv():
 with open('C:\\data.csv', mode='a', newline='') as file:
 writer = csv.writer(file, delimiter=',', quotechar="")
 writer.writerow(['4', 'Devansh', 'Arts', '404'])

print("Press-1 to Read Data and Press-2 to Write data: ")
a=int(input())
if a==1:
 readcsv()
elif a==2:
 writecsv()
else:
 print("Invalid value")
```

### OUTPUT:

```
Press-1 to Read Data and Press-2 to Write data: 1
['Roll No.', 'Name of student', 'stream', 'Marks']
['1', 'Anil', 'Arts', '426']
['2', 'Sujata', 'Science', '412']
['3', 'Shivani', 'Commerce', '448']
['4', 'Devansh', 'Arts', '404']
```

**Program-2**:- Create a CSV file by entering user-id and password, read and search the password for given user-id.

### CODE:

```
import csv
with open("user_info.csv", "w") as obj:
 fileobj = csv.writer(obj)
 fileobj.writerow(["User Id", "password"])
 while(True):
 user_id = input("enter id: ")
 password = input("enter password: ")
 record = [user_id, password]
 fileobj.writerow(record)
 x = input("press Y/y to continue and N/n to terminate the program\n")
 if x in "Nn":
 break
 elif x in "Yy":
 continue
with open("user_info.csv", "r") as obj2:
 fileobj2 = csv.reader(obj2)
 given = input("enter the user id to be searched\n")
 for i in fileobj2:
 next(fileobj2)
 # print(i,given)
 if i[0] == given:
 print(i[1])
 break
```

#### **OUTPUT:**

```
enter id: cbse
enter password: 123
press Y/y to continue and N/n to terminate the program
y
enter id: computer_science
enter password: python
press Y/y to continue and N/n to terminate the program
n
enter the user id to be searched
cbse
123
>>>
```

#### **Sample Questions & Answers**

1. What does the acronym CSV stand for in its full form?

- a. Common Separated Value
- b. Comma System Value
- c. Comma Separated Value
- d. Common System Vault

Ans: c. Comma Separated Value

2. What is the default delimiter of a CSV file

- a. Tab
- b. Comma
- c. Semicolon
- d. Space

Ans: b. Comma

3. In regards to separated value files such as .csv and .tsv, what is the delimiter?

- a. Any character such as the comma (,) or tab (\t) that is used to separate the **row** data
- b. Anywhere the comma (,) character is used in the file
- c. Delimiters are not used in separated value files
- d. Any character such as the comma (,) or tab (\t) that is used to separate the **column** data.

Ans: d. Any character such as the comma (,) or tab (\t) that is used to separate the **column** data.

4. In separated value files such as .csv and .tsv, what does the first row in the file typically contain?

- a. The source of the data
- b. The author of the table data
- c. Notes about the table data
- d. The column names of the data

Ans: d. The column names of the data

5. Assume you have a file object my\_data which has properly opened a separated value file that uses the tab character (\t) as the delimiter.

What is the proper way to open the file using the Python csv module and assign it to the variable csv\_reader?

Assume that csv has already been imported.

- a. csv\_reader = csv.tab\_reader(my\_data)
- b. csv\_reader = csv.reader(my\_data)
- c. csv\_reader = csv.reader(my\_data, tab\_delimited=True)
- d. csv\_reader = csv.reader(my\_data, delimiter='\t')

Ans: d. csv\_reader = csv.reader(my\_data, delimiter='\t')

6. When iterating over an object returned from csv.reader(), what is returned with each iteration?

For example, given the following code block that assumes csv\_reader is an object

returned from csv.reader(), what would be printed to the console with each iteration?

```
for item in csv_reader:
 print(item)
```

- a. The individual value data that is separated by the delimiter
- b. The row data as a list
- c. The column data as a list
- d. The full line of the file as a string

Ans: b. The row data as a list.

## 7. Case study Question-

Neha is making software on “Items & their prices” in which various records are to be stored/retrieved in STORE.CSV data file. It consists some records (Item & Price). She has written the following code in python. As a programmer, you have to help her to successfully execute the program.

```
import _____ # Statement-1
def AddItem(Item, Price) _____
 f=open("STORE.CSV", _____) # Statement-2
 fw=csv.writer(f)
 fw.writerow([Item, Price]) # Statement-3

def ShowRecord():
 with open("STORE.CSV","r") as NI:
 NewItem=csv._____ (NI) # Statement-4
 for rec in NewItem:
 print(rec[0], "#", rec[1])

#main-code
AddItem("Sugar", 38.00)
AddItem("Rice", 48.50)
ShowRecord() # Statement-5
```

Q1. Which module should be imported in Statement-1.

- A. pickle
- B. csv
- C. file
- D. text

Ans: B. csv

Q2. Which file mode to be passed to add new record in Statement-3.

- A. w+
- B. w
- C. wb
- D. a

Ans: D. a

Q3. What should be written in Statement-4 to close the file?

- A. close()
- B. fw.close()
- C. f.close()
- D. csv.close()

Ans: C. f.close()

Q4. Which function to be used in Statement-5 to read the data from a csv

file.

- A. read()      B. readline()      C. readlines()      D. reader()

Ans: D. reader()

Q5.      Output after executing Statement-6 will be -

- A. ("Sugar", "38.0")  
("Rice", "48.50")
- B. Sugar 38.0  
Rice 48.0
- C. Sugar, 38.0  
Rice, 48.50
- D. Sugar # 38.0  
Rice # 48.50

Ans: D.      Sugar # 38.0  
Rice # 48.50

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

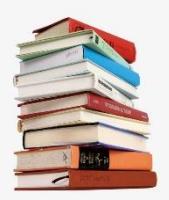
### **Introduction :**

**Data structures :** A data structure is a named group of data of different data types which is stored in a specific way and can be processed as a single unit. It has well-defined operations, behavior and properties.

**Stack :** It is a linear structure implemented in LIFO (Last In First Out) manner where insertions and deletions are restricted to occur only at one end – stack's top.

**LIFO** means element last inserted would be the first one to be deleted.  
For example, a pile of books, a stack of coins, where you can remove only the top book or the coin placed at the top (Fig.1).

Fig.1 : Pile of Books / Stack  
Some of the  
are:



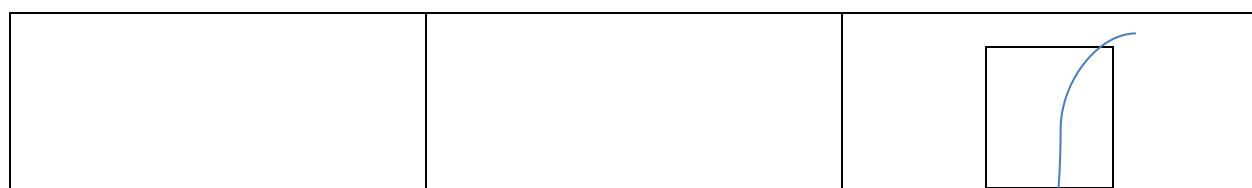
of Coins  
applications of Stack in real life

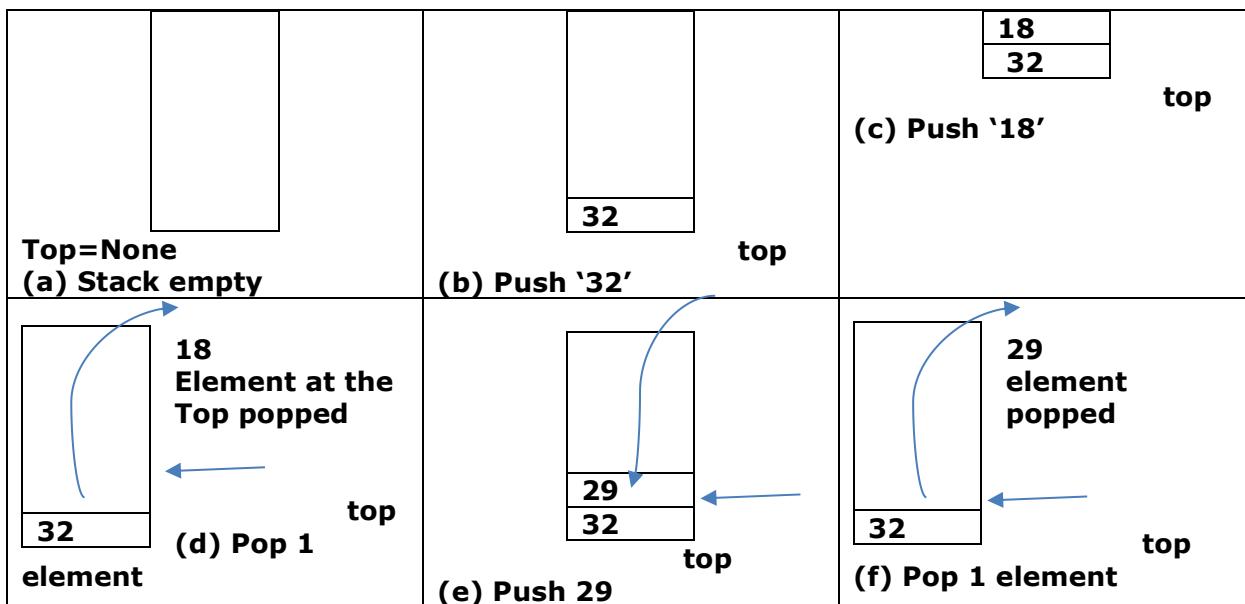
- Pile of clothes in an almirah.
- Multiple chairs in a vertical pile
- Bangles worn on wrist
- Pile of boxes of eatables in a pantry or on a kitchen shelf

Stack follows these rules :

- 1) Data can only be removed from the top (pop). The removal of element from a stack is called **POP operation**.
- 2) A new data element can only be added to the top of the stack (push). The insertion of element in a stack is called **PUSH operation**.

**Following figure illustrates the operations on a stack.**





Stack is a dynamic data structure as it grows (with an increase in the number of elements) or shrinks (with a decrease in the number of elements). A static data structure is the one that has fixed size.

## Implementation of Stack using List

The basic operations performed on the Stack are:

1. Creating a Stack
2. PUSH Operation / Adding Elements to a Stack
3. Pop Operation / Deleting Elements from a Stack
4. Traversal / Displaying a Stack.

Checking the Status of Stack : Before implementing these operations, it is important to check Stack status.

### 1. Creating a Stack

To create an empty Stack, use inbuilt function `list()` or `[]` as per the syntax given below:

Syntax :

`Stack=list()`

`Stack=[]`

In order to access the elements in a Stack, index values are used. Thus, the first element in the Stack will be `Stack[0]`, and so on.

### 2. PUSH Operation adding Elements into a Stack

The list function `append()` is used to add elements into the Stack.

Syntax:

`Stack.append(x)`

Here, `x` is the element to be inserted into the Stack and is inserted at the end of the list implemented as a Stack.

**peek()** : `peek()` function is used to get the most recent value of stack, i.e., top element of the stack without removing it or value at the top. It will throw an exception if the stack is empty or null.

### 3. Pop Operation/Deleting an Element from a Stack

The list function `pop()` is used to pop / remove/delete elements from a Stack.

Syntax :

`Stack.pop()`

It returns the popped / deleted elements from the list.

#### **4. Traversing /displaying Stack Elements**

In this process the elements from the top position, i.e., last inserted element, get displayed and processed in the reverse order.

#### **Applications of Stacks**

1. Reversing a Word/Line.
2. The compilers use Stacks to store the previous state of a program when a function is called or during recursion.
3. Backtracking : Backtracking is a form of recursion. But it involves choosing only one option out of possibilities. It is used in large number of puzzles like Sudoku and in optimization problems such as knapsack.
4. Undo Mechanism in text Editors.

#### **Practical Implementation-1**

**Write a Python program to implement all basic operations of a Stack, such as adding element, removing element and displaying the Stack elements using lists.**

```
s=[]
c="y"
while (c=="y"):
 print("1. PUSH")
 print("2. POP")

 print("3. Display")
 choice=int(input("Enter your choice: "))
 if (choice==1):
 a = input("enter any number :")
 s.append(a)
 elif (choice==2):
 if (s==[]):
 print("Stack Empty")
 else:
 print("Deleted element is : ",s.pop())
 elif (choice==3):
 l=len(s)
 for i in range(l-1,-1,-1):
 print(s[i])
 else:
 print("Wrong Input")
c=input("Do you want to continue or not?")
```

#### **Practical Implementation – 2**

**Write a program to display unique vowels present in the given word using Stack.**

```
vowels=['a','e','i','o','u']
word=input("Enter the word to search for vowels :")
Stack=[]
for letter in word:
 if letter in vowels:
```

```

if letter not in Stack:
 Stack.append(letter)
print(Stack)
print("The number of different vowels present in ",word," is ",len(Stack))

```

### Practical Implementation – 3

**Write a program in Python to implement the following operation on Stack containing employee code and name. 1. PUSH 2. POP 3. DISPLAY**

```

employee = []
choice = 'y'
while (choice=='y' or choice=='Y'):
 print("1. PUSH")
 print("2. POP")
 print("3. DISPLAY STACK")
 choice1 = int(input("Enter any choice: "))
 if(choice1==1):
 e_id= input('Enter employee code ')
 ename=input('Enter name ')
 emp = (e_id, ename)
 employee.append(emp)
 elif(choice1==2):
 if len(employee)==0:
 print('Stack empty')
 else:
 e_id, ename = employee.pop()
 print('Deleted element is ',e_id, ename)
 elif(choice1==3):
 top = len(employee)
 while top>0:
 print(employee[top-1])
 top = top -1
 else:
 print('Wrong Input')
 choice = input('Do you want to continue? y or n: ')

```

### MCQ / Short Answer Type Question – 1 Mark

- Choose the correct output for the following stack operation(\* top position)  
Push(5) Push(8) Pop() Push(2) Push(5) Pop() Push(1)  
(a) 8 5 2 5 1\*      (b) 8 5 5 2 1\*      (c) 2 5 5 1\*      (d) 5 2 1\*
- Which list method can be used to perform Push operation in a stack implemented by list?  
(a) append()      (b) extend()      (c) push()      (d) insert()
- Which list method can be used to perform Pop operation in a stack implemented by list?  
(a) pop()      (b) pop(1)      (c) remove()      (d) pop(0)
- Consider the following operation performed on a stack of size 3, What will be the output? (\* top position)  
Push(10) Push(20) Push(30) Pop() Push(40) Push(50)  
(a) overflow      (b) underflow      (c) 10 20 30 40 50\*      (d) 10 20 40 50\*
- What is the situation called when an insertion is attempted in a full Stack?

6. What is the situation called when read/deletion is attempted in an empty Stack?
7. LIFO data structure is?
8. Name the operation for insertion in a stack.
9. Name the operation for deletion from a stack.
10. Define Stack.

### **Short Long Answer Type Question – 2 Marks**

1. Write some applications of stack.
2. Consider STACK=['a','b','c','d']. Write the STACK content after each operations:  
a) STACK.pop() b) STACK.append('e') c) STACK.append('f') d) STACK.pop()
3. Write a program to implement a stack for the students(studentno, name). Just implement Push.
4. Given a bounded stack of capacity 4 which is initially empty, write the stack content after each steps:  
a) Push '1' b) Push '2' c) Push '3' d) Push '4' e) Pop f) Pop  
g) Push('5') h)Pop i)Pop j)Pop

5. What is Stack? What basic operations can be performed on them?

6. Find the output of the following code:

```
result=0
numberList=[10,20,30]
numberList.append(40)
result=result+ numberList.pop()
result=result+ numberList.pop()
print(result)
print(numberList)
```

7. Find the output of the following code:

```
answer=[]
output=''
answer.append('T')
answer.append('A')
answer.append('M')
ch=answer.pop()
output=output+ch
ch=answer.pop()
output=output+ch
ch=answer.pop()
output=output+ch
print("Result=",output)
```

8. What is backtracking? What is the use of it?

9. Find the output of the following code:

```
stack =["Amar", "Akbar", "Anthony"]
stack.append("Ram")
stack.append("Iqbal")
print(stack)
print(stack.pop())
print(stack)
print(stack.pop())
print(stack)
```

10. What is the use of peek() function?

### **Long Answer Type Question – 3 Marks**

1. Write a function in Python **PUSH\_IN(L)**, where L is a list of numbers. From this list, push all even numbers into a stack which is implemented by using another list.

2. Write a function in Python **POP\_OUT(Stk)**, where Stk is a stack implemented by a list of numbers. The function returns the value which is deleted/popped from the stack.

3. Write a function in Python PUSH(Arr), where Arr is a list of numbers. From this list push all numbers divisible by 5 into a stack implemented by using a list. Display the stack if it has at least one element, otherwise display appropriate error message.

4. Write a function in Python POP(Arr), where Arr is a stack implemented by a list of numbers. The function returns the value deleted from the stack.

5. Write a function in python, PushEl(e) to add a new element and PopEl(e) to delete a element from a List ,considering them to act as push and pop operations of the Stack data structure .

6. Write a function POP(Book) in Python to delete a Book from a list of Book titles, considering it to act as a pop operation of the Stack data structure.

7. Julie has created a dictionary containing names and marks as key value pairs of 6 students. Write a program, with separate user defined functions to perform the following operations:

- Push the keys (name of the student) of the dictionary into a stack, where the corresponding value (marks) is greater than 75.
- Pop and display the content of the stack.

For example: If the sample content of the dictionary is as follows:

R={"OM":76, "JAI":45, "BOB":89, "ALI":65, "ANU":90, "TOM":82}

The output from the program should be: TOM ANU BOB OM

8. Alam has a list containing 10 integers. You need to help him create a program with separate user defined functions to perform the following operations based on this list.

- Traverse the content of the list and push the even numbers into a stack.
- Pop and display the content of the stack.

For Example: If the sample Content of the list is as follows:

N=[12, 13, 34, 56, 21, 79, 98, 22, 35, 38]

Sample Output of the code should be: 38 22 98 56 34 12

9. Coach Abhishek stores the races and participants in a dictionary. Write a program, with separate user defined functions to perform the following operations:

Push the names of the participants of the dictionary onto a stack, where the distance is more than 100.

Pop and display the content of the stack.

For example: If the sample content of the dictionary is as follows:

```
Races = {100:'Varnika', 200 : 'Jugal', 400:'Kushal', 800:'Minisha'}
```

The output from the program should be: Minisha Kushal Jugal

10. Reva has created a dictionary containing Product names and prices as key value pairs of 4 products. Write a user defined function for the following:

- PRODPUSH() which takes a list as stack and the above dictionary as the parameters.

Push the keys (Pname of the product) of the dictionary into a stack, where the corresponding price of the products is less than 6000. Also write the statement to call the above function.

For example: If Reva has created the dictionary is as follows:

```
Product = {"TV":10000, "MOBILE":4500, "PC":12500, "FURNITURE":5500}
```

The output from the program should be: [ 'FURNITURE', 'MOBILE' ]

### **Case Based Questions – 5 Marks**

1. Based on the below given code, Write answer to the following questions i to v

```
#function definition
```

```
def push(colour,n):
```

```
-----# statement1
```

```
#function calling
```

```
colour = []
```

```
c = ['red', 'blue', 'violet', 'white', 'black']
```

```
for i in range(0, len(c), 2):
```

```
.....# statement2
```

```
print(colour[::-1])
```

```
#function definition
```

```
def pop(colour):
```

```
if#statement 3
```

```
 return "underflow"
```

```
else:
```

```
 return#statement 4
```

```
#function calling
```

```
for i in range(len(colour) + 1):
```

```
 print(.....)#statement 5
```

(i) Identify the suitable code for statement 1?

a) colour.insert(len(colour),n)

b) colour.append(len(colour),n)

c) colour.append()

d) colour.extend()

(ii) Identify the suitable code for statement 2?

a) push(colour,c[i]) b) push(colour) c) push(c[i]) d) push(colour,i)

(iii) Identify the suitable code for statement 3?

- a) colour==[]: b) colour.isEmpty(): c) len(colour)=0: d) None of the above  
 (iv) Fill in the statement to delete an element from the stack?  
 a) colour.pop(1) b) colour.pop() c) del colour[1] d) colour.delete(1)  
 (v) Fill the statement 5, to call pop function  
 a) pop(c)b) pop(colour) c) call pop(colour) d) def pop(colour)

2. Based on the below given code, Write answer to the following questions i to v.

```
course_dict = {'BSC':8000, 'MSC':12000, 'BCA':15000, 'PGDCA':9000,
```

```
'MCA':30000, 'MBA':50000 }
```

```
STACK = []
```

```
def NEWDATA():
```

```
 course = input('Enter course name :')
```

```
 fee = int(input('Enter fee '))
```

```
 #statement 1
```

```
 print('Data saved successfully ')
```

```
 print(course_dict)
```

```
def PUSH():
```

```
 for key in course_dict:
```

```
 if _____ > 10000: #statement 2
```

```
 STACK._____ #statement 3
```

```
def POP():
```

```
 if _____ ==0: #statement 4
```

```
 print('STACK EMPTY, No course with fee>10000')
```

```
 else:
```

```
 print('course fee more than 10000')
```

```
 for top in range(len(STACK), 0, -1):
```

```
 print(_____) #statement 5
```

```
NEWDATA()
```

```
PUSH()
```

```
POP()
```

Consider a dictionary with keys as course name and fee as value. Write a program to push course name in stack where fee is more than 10000. Pop and display contents of stack on the screen

- Write a code to add the fee and course name to course\_dict.
- Complete the code to compare the fee of the given key(course).
- Write a code to insert the element in the Stack.
- Write a code to find the length of the Stack.
- Write a code to delete the element of the Stack..

### **ANSWER KEY**

#### **MCQ / Short Answer Type Question – 1 Mark**

- |              |                 |              |                 |
|--------------|-----------------|--------------|-----------------|
| 1.(d) 5 2 1* | 2. (a) append() | 3. (a) pop() | 4. (a) overflow |
| 5. Overflow  | 6. Underflow    | 7. STACK     | 8. PUSH         |
| 9. POP       |                 |              |                 |

10. It is a linear structure implemented in LIFO (Last In First Out) manner where insertions and deletions are restricted to occur only at one end – stack's top.

#### **Short Long Answer Type Question – 2 Marks**

- Reversing a string, compilers use stack to store previous state of program, undo mechanism in text editors and backtracking.
- a) ['a', 'b', 'c']  
 b) ['a', 'b', 'c','e']  
 c) ['a', 'b', 'c','e','f']

d) ['a', 'b', 'c', 'e']

3. Ans: Program for push operation in a stack

stk=[]

top=-1

def PUSH(stk,student):

    stk.append(student)

    top=len(stk)-1

sno=int(input("Enter student No:"))

sn=input("Enter student Name:")

data=[sno,sn]

PUSH(stk,data)

4. a) stack=['1']      b) stack=['1','2'] c) stack=['1','2','3']

d) stack=['1','2','3','4']

e) stack=['1','2','3']      f) stack=['1','2']      g) stack=['1','2','5']

h) stack=['1','2']      i) stack=['1']      j) stack=[]

5. Stack is a linear structure implemented in LIFO (Last In First Out) manner where insertions and deletions are restricted to occur only at one end – stack's top.

Following basic operations can be performed on Stack.

(i) PUSH, i.e., insertion of element in the Stack.

(ii) POP, i.e., deletion of an element from the Stack.

(iii) Traversal , i.e., displaying all the elements of the Stack.

6. 70

[10,20]

7. Result=MAT

8. Backtracking is a form of recursion. But it involves choosing only one option out of possibilities. It is used in large number of puzzles like Sudoku and in optimization problems such as knapsack.

9.

['Amar', 'Akbar', 'Anthony', 'Ram', 'Iqbal']

Iqbal

['Amar', 'Akbar', 'Anthony', 'Ram']

Ram

['Amar', 'Akbar', 'Anthony']

10. peek() function is used to get the most recent value of stack, i.e., top element of the stack without removing it or value at the top. It will throw an exception if the stack is empty or null.

### Long Answer Type Question – 3 Marks

1. top=-1

stk=[]

def PUSH\_IN(L):      # Allow additions to the stack

    for i in L:

        if i%2==0:

            stk.append(i)

            top=len(stk)-1

2.

def isEmpty(stk):      # checks whether the stack is empty or not

```

if stk==[]:
 return True
else:
 return False
def POP_OUT(stk):
 if isEmpty(stk): # verifies whether the stack is empty or not
 print("Stack Underflow")
 else: # Allow deletions from the stack
 item=stk.pop()
 if len(stk)==0:
 top=-1
 else:
 top=len(stk)
 return item

```

### **3.**

```

def PUSH(Arr,value):
 s=[]
 for x in range(0,len(Arr)):
 if Arr[x]%5==0:
 s.append(Arr[x])
 if len(s)==0:
 print("Empty Stack")
 else:
 print(s)

```

### **4.**

```

def popStack(st) : # If stack is empty
 if len(st)==0:
 print("Underflow")
 else:
 L = len(st)
 val=st[L-1]
 print(val)
 st.pop(L-1)

```

### **5.**

```

def PushEl(element):
 a=int(input("enter package title : "))

```

```

 element.append(a)
def PopEl(element):
 if (element==[]):
 print("Stack empty")
 else:
 print ("Deleted element:", element.pop())

```

**6.**

```

def POP(Book):
 if (Book ==[]):
 print("Stack empty")
 else:
 print("Deleted element :")
 Book.pop()

```

**7.**

```

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

```

**8.**

```
N=[12, 13, 34, 56, 21, 79, 98, 22, 35, 38]
```

```
def PUSH(S,N):
 S.append(N)
```

```
def POP(S):
 if S!=[]:
```

```
 return S.pop()
```

```
 else:
```

```
 return None
```

```
ST=[]
```

```
for k in N:
```

```
 if k%2==0:
```

```
 PUSH(ST,k)
```

```
while True:
```

```
 if ST!=[]:
```

```
 print(POP(ST),end=" ")
```

```
 else:
```

```
break
```

**9.**

```
Races ={100:'Varnika',200:'Jugal', 400:'Kushal',800:'Minisha'}
def PUSH(Stk,N):
 Stk.append(N)
def POP(Stk):
 if Stk!=[]:
 return Stk.pop()
 else:
 return None
stack1=[]
for i in Races:
 if i>=200:
 PUSH(stack1,Races[i])
while True:
 if stack1!=[]:
 print(POP(stack1),end=" ")
 else:
 break
```

**10.**

```
Product={"TV":10000, "MOBILE":4500, "PC":12500, "FURNITURE":5500}
stack=[]
def PRODPUSH(stack, P):
 for k in Product:
 if P[k]<6000:
 stack.append(k)

PRODPUSH(stack, Product)
print(stack)
```

**Case Based Questions – 5 Marks**

1. (i). a colour.insert(len(colour),n)
- (ii). a push(colour,c[i])
- (iii).a colour==[]
- (iv). b colour.pop()
- (v). b pop(colour)

2. i) course\_dict[course] = fee
- ii) course\_dict[key]
- iii) append(key)
- iv) len(STACK)
- v) STACK.pop()

**Unit II: Computer Networks**

- a. Evolution of networking: introduction to computer networks, evolution of networking (ARPANET, NSFNET, INTERNET)
- b. Data communication terminologies: concept of communication, components of data communication (sender, receiver, message, communication media,

protocols), measuring capacity of communication media (bandwidth, data transfer rate), IP address, switching techniques (Circuit switching, Packet switching)

### **Point wise notes/Key Points/ Summary**

A system of interconnected computers and computerized peripherals such as printers is called computer network. This interconnection among computers facilitates information sharing among them. Computers may connect to each other by either wired or wireless media.

### **Network Applications**

Computer systems and peripherals are connected to form a network. They provide numerous advantages:

- Resource sharing such as printers and storage devices
- Exchange of information by means of e-Mails and FTP
- Information sharing by using Web or Internet
- Interaction with other users using dynamic web pages
- IP phones
- Video conferences
- Parallel computing
- Instant messaging

**ARPANET**, in full Advanced Research Projects Agency Network, experimental computer network that was the forerunner of the Internet. The Advanced Research Projects Agency (ARPA), an arm of the U.S. Defense Department, funded the development of the Advanced Research Projects Agency Network (ARPANET) in the late 1960s. Its initial purpose was to link computers at Pentagon-funded research institutions over telephone lines.

**NSFNET**: A variety of regional research and education networks, supported in part by NSF, were connected to the NSFNET backbone, thus extending the Internet's reach throughout the United States.

### **INTERNET**

The Internet is a collection of computers connected by network cables or through satellite links. Rather than connecting every computer on the Internet with every other computer, individual computers in an organization are normally connected in a local area network (LAN). One node on this local area network is physically connected to the Internet. So, the Internet is a network of networks.

### **PROTOCOL**

A network protocol is an established set of rules that determine how data is transmitted between different devices in the same network. Essentially, it allows connected devices to communicate with each other, regardless of any differences in their internal processes, structure or design. Network protocols are the reason you can easily communicate with people all over the world, and thus play a critical role in modern digital communications.

An IP address is a unique address that identifies a device on the internet or a local network. IP stands for "Internet Protocol," which is the set of rules governing the format of data sent via the internet or local network.

## **Switching Techniques**

**Circuit switching** is a switching method where an end-to-end path is created between two stations within a network before starting the data transfer.

**Packet switching** is the transfer of small pieces of data across various networks. These data chunks or “packets” allow for faster, more efficient data transfer. Often, when a user sends a file across a network, it gets transferred in smaller data packets, not in one piece.

### **One marks Questions**

- 1) ARPANET stands for \_\_\_\_\_.
- 2) Give any two advantages of computer network.
- 3) Give any two disadvantages of computer network.
- 4) NFSNET stands for \_\_\_\_\_.
- 5) TCP / IP stands for \_\_\_\_\_.
- 6) Which one is not a part of data communication.
  - a. Sender
  - b. Receiver
  - c. Message
  - d. Protocol
  - e. None of these
- 7) ISP stands for \_\_\_\_\_.
- 8) What is a protocol?
- 9) LAN stands for \_\_\_\_\_.
- 10) Telephone network is an example of \_\_\_\_\_.
  - a. Circuit Switching
  - b. Packet Switching
  - c. Message Switching
  - d. All of these
- 11) 1 kbps = \_\_\_\_\_ bps.
- 12) Modem stands for \_\_\_\_\_.
- 13) Circuit switching requires a dedicated path. (True / False)
- 14) Packet switching is more efficient because the cost of the link is shared by many users. (True / False)
- 15) Out of the following which is the fastest wired medium of communication.  
Infrared, Coaxial cable, Ethernet Cable, Microwave, Optical Fiber.
- 16) Out of the following which is the fastest wireless medium of communication.  
Infrared, Coaxial cable, Ethernet Cable, Microwave, Optical Fiber.
- 17) Name the protocol that is used to send emails.
- 18) Which communication medium is to be suggested for very effective and fast communication in guided medium?

### **Two marks Questions**

- 1) Define a network. What is its need?
- 2) What is a network? What are its goals and applications.
- 3) Write any two advantages and disadvantages of network.
- 4) Rearrange the following terms in increasing order of data transfer rates.  
Gbps, Kbps, Mbps, Tbps, Bps, Kbps
- 5) What is a protocol? Name some commonly used protocols.
- 6) How many switching techniques are there? Explain any one with example.
- 7) Define baud, bps and Bps. How are these interlinked?
- 8) Why we used HTTP protocol.
- 9) Why packet switching is more efficient than circuit switching?
- 10) Rakshana, a class Xi student, has just started understanding the basics of internet and web

technologies. She is a bit confused about the terms “World Wide Web” and “Internet”. Help her to understand both the terms with the help of suitable example of each.

**11) Define the following terms.**

- a. Baud
- b. Communication Channel

**12) Computer network is made from computers only, Justify the sentence.**

**13) Protocols are more important in the computer network because they help to control computer network. Is it true or false? Give the proper reason for true or false.**

**Three marks Questions**

- 1) Describe the role of ARPANET and NSFNET in evolution of network.
- 2) Differentiate the Analog and Digital modes of data transmission using real time applications.
- 3) List and explain the components of Data Communication.
- 4) Neha is confused in the working process of landline Telephone Network and Mobile Network. Kindly help her to explain Circuit switching and Packet switching with proper examples.
- 5) Where we can used packet switching? Explain the packet switching with example.
- 6) What is TCP/IP? How FTP is different from HTTP?
- 7) Expand the below terms regarding protocol. (Any 6)

|         |           |        |
|---------|-----------|--------|
| a. HTTP | d. IMAP   | g. POP |
| b. PPP  | e. FTP    | h. IP  |
| c. SMAP | f. TCP/IP |        |
- 8) Define the below terminology of networks.
  - a. Server
  - b. IP Address
  - c. Domain Name

#### **Case Study 1**

Kiran has started a business all over India and he wants to establish a network among its offices and branches. Help him to choose various communication media as per requirement.

**Head Office: Ahmedabad, Gujarat, 10 Computers**

**Branch 1: Delhi, 50 Computers**

**Branch 2: Mumbai, 100 Computers**

**Branch 3: Kolkata, 100 Computers**

- 1) Kiran wants high speed network in Branch 1, Delhi. Which cable he has to used to provide maximum speed among computers. (1 Marks)
- 2) Kiran has low budget in Branch 2, Mumbai. Which cable he has to use to keep low cost network establishment. (1 Marks)
- 3) Kiran does not know about guided or unguided media. Kindly help him to classify the below media into Guided and Unguided media. (2 Marks)
  - a. Fiber Optic
  - b. Radio wave
  - c. Co-axial
  - d. Microwave
- 4) Kiran wants to connect Head office with all branches. Which communication media he has to used to connect head office with all branches. (1 Marks)

#### **Case Study 2**

Jinesh is studying in one of the popular Hardware and Networking institute. He already learned Network Topology, Types of Network and Communication media. He has just started to learn about protocol. Kindly help him to learn protocols with below question answers.

- 1) Which protocol is used to send or receive emails? (1 Marks)
- 2) Which protocol is used to surf the web? (1 Marks)
- 3) Which protocol is used to surf the web securely? (1 Marks)
- 4) Which protocol is used to send or receive the files? (1 Marks)

5) Expand the POP protocol.

(1 Marks)

#### Summary:

A system of interconnected computers and computerized peripherals such as printers is called computer network. This interconnection among computers facilitates information sharing among them. Computers may connect to each other by either wired or wireless media. To connect we require protocols, Switching technology, communication media and structure of computer network. We can use computer network in Resource sharing such as printers and storage devices, Information sharing by using Web or Internet and Interaction with other users using dynamic web page.

#### Answers :

##### One Marks questions:

- |                                                              |                   |
|--------------------------------------------------------------|-------------------|
| 1) Advanced Research Project Agency Network                  |                   |
| 2) Resource sharing, Reliability of data or any related it.  |                   |
| 3) Cost of setup, Threat to Data Security or any related it. |                   |
| 4) National Science Federation Network                       |                   |
| 5) Transmission Control Protocol over Internet Protocol      |                   |
| 6) E. none of these                                          | 13) True          |
| 7) Internet Service Provider                                 | 14) True          |
| 8) 1 mark for correct definition                             | 15) Optical Fiber |
| 9) Local Area Network                                        | 16) Microwave     |
| 10) A. Circuit Switching                                     | 17) IMAP, POP     |
| 11) 1024                                                     | 18) Optical Fiber |
| 12) Modulator Demodulator                                    |                   |

#### Case Study 1

- 1) Fiber Optic Cable
- 2) Co-axial cable or Twisted Pair Cable
- 3)
  - a. Guided Media
    - i. Fiber Optic
    - ii. Co-axial
  - b. Unguided Media
    - i. Radio Wave
    - ii. Micro Wave
- 4) Satellite Link

#### Case Study 2

- |              |                         |
|--------------|-------------------------|
| 1) IMAP, POP | 4) FTP                  |
| 2) HTTP      | 5) Post Office Protocol |
| 3) HTTPS     |                         |

## TRANSMISSION MEDIA ( 1 Mark questions )

1. How many type of transmission media are there? List the types.

Ans: Two types: Wired and Wireless.

2. List wired media.

Ans: Twisted Pair, Coaxial, Fibre Optics cable.

3. List Wireless Media.

Ans: Bluetooth, Infrared, Wi-Fi, Radio waves, Satellite Link.

4. What is the maximum distance to which twisted pair cable can send data?

Ans: 100m

5. What is the maximum distance to which co-axial cable can send data?

Ans: 500m

6. The network type connected through Bluetooth is \_\_\_\_\_

Ans: PAN

7. Satellite link is the example of which type of network?

Ans: WAN

8. KVKV co. has established a network which spreads in Jaipur city. Which type of network KVKV co. has established among LAN/MAN/WAN?

Ans: MAN.

9. ISRO (India) wants to establish a direct connection with NASA (US) and ESA (Europe). Which connection media can be used?

Ans: Satellite link or Fibre optics.

10. Wired media is also known as,

Ans: Guided media, Physical media, Conducted media.

11. Which connector is used to plug the twisted pair cable with device?

Ans: RJ45 connector

12. What is bandwidth of cat-6 twisted pair cable?

Ans: 1 Gbps.

## **2 marks questions**

1. Write two features of LAN

- Can spread in a room or small area.
- Established with twisted pair cable or wi-fi.

2. Write the two features of PAN.

- Established with USB cable, Bluetooth or infrared.
- Very small area upto few feet.
- Ad-hoc network. Devices can join and leave anytime.
- Less secure.

3. Write any two advantages of co-axial cable.

- High bandwidth.
- Can be used for upto 500 mtr distance.

4. What is RJ 45 connector?

- Used with twisted pair cable
- Full form is Registered Jack
- Plugin to network devices
- Primarily for LANs

5. Write features of Tree topology.

- No cycle
- Connection to all PCs
  - Either direct or through other PCs
- Can say, it is a group of stars
- A fault breakdowns the network

6. Write features of Mesh topology.

- Connected in many to many concept.
- Fastest connection topology
- Very costly
- Lots of cable required
- Robust network
- High speed

7. What is peer to peer network?

- Only two devices

- Can have separate wires
  - Easiest
  - More secure than others
8. What is Ethernet Card?
- A hardware for connecting nodes
  - Transmit data to network from computer
  - Receive data from network to computer
  - Can connect one cable or more
9. What is repeater?
- Repeater is a network device used to boost the signal.
  - Used to boost the signals to send for long distance.
  - Also known as signal booster.
  - Its task is to amplify the signal.
10. Explain networking Hub.
- A Hub is a networking device that allows you to connect multiple PCs to a single network.
  - It is used to connect segments of a LAN.
  - A hub stores various ports, so when a packet arrives at one port, it is copied to various other ports.
  - Hub works as a common connection point for devices in a network.

### **3 mark questions.**

1. Write three disadvantages of co-axial cable.

- Less flexible.
- More expensive than twisted pair.
- Not compatible with modern cables.
- Difficult to install.

4. What is the structure of fibre optics cable?

Core – It is the innermost portion of an optical fiber through which light propagates. It is cylindrical in shape and made up of a flexible glass of high refractive index. The diameter of the core of a single mode fiber is 8 – 10 µm while multimode fibers are 50 µm in diameter. It is also called the optical waveguide since it is the main channel through which light signals are transmitted.

Cladding – The core is surrounded by a glass cladding. The glass of cladding has a lower refractive index than the core. This enables total internal reflection of light waves in the core, and eventually propagation of light waves within the core.

Outer Coating or Jacket – The outer jacket is a thin plastic sheath or coating that is opaque to light. It prevents light rays from outside to enter the optical fiber. Fibers are typically bundled together, where each bundle is covered by a protective outer sheath that prevents the fibers from physical damage.

5. How data are transmitted through fibre optics?

- The binary data is converted into light signals/pulses.
- The light pulses are passed into fibre optics core through LED.
- The light passes through glass core of fibre optics and reaches at another end.
- The receiver at another end receives light pulses and converts them back to binary data.

6. Write advantages and disadvantages of fibre optics.

- Advantages
  - Free from EMI coz it has no electrical signals

- Secure
- High speed
- Very long distance
- Most efficient cable for computer networks
- Disadvantages
  - Costly
  - Fragile
  - Complicated installation and maintenance.

7. Explain Infrared.

- A very short distance transmission.
- Used the Infrared Light using LED.
- Sends the IR light pulses(codes), which is received by IR receiver.
- Connection can be within 100 ft (30.5mtrs)
- 10 Mbps speed.
- Insensitive to radio-frequency interference
- Can be degraded by bright light.

8. Write short note about Bluetooth

- Ad-hoc network. Means established when required and with which devices is to be connected with.
- Can connect limited number of devices.
- Devices ex: Mouse, keyboard, headsets, printers etc. uses Bluetooth to connect wirelessly.
- It uses 2.4 GHz frequency.
- Max distance for proper connection is 10 mtrs.
- Transfer rate is 50 Mbps of Bluetooth 5.0

9. Write short note about Wi-Fi

- Wireless Fidelity
- High-speed, low-cost access to internet
- Distance 100 meters=328 ft
- Faster than Bluetooth
- Wi-Fi access points are deployed in Dept stores, railway stations, airports, colleges, hotels etc.
- IEEE 802.11 standard stands for Wi-Fi.

10. Write features of radio waves, its advantages and disadvantages.

- Radio Waves
  - Electromagnetic wave/sine waves
  - Wavelength 0.5cm to 30000m
  - Transmitter, Receiver
  - License required to use
- Benefits
  - Mobility
  - Cheaper than cables
  - Ease of communication over difficult terrain
  - Land acquisition not required
- Disadvantages
  - Insecure
  - Susceptible to weather

11. Write short note about router.

- Connects two or more networks
- Router contains the processor and memory
- It can process the network data for filtering, validating
- It can store data for traffic management
- Router has on-board OS and firmware

- It stores the addresses of all connected nodes to it
- It can manage flow of data and path of data

12. Write short note about switch.

- Switch is an electronic device
- Connects computers together
- It receives and forward data to particular node
- It uses hardware addresses (MAC)
- It has many LAN ports
- LAN cable can be attached to any port
- Its size can be few inches
- If Switch speed is 100Mbps or more, it is Ethernet switch

13. What is modem?

- Its task is Modulation, Demodulation of signals.
- Converts analog to digital and digital to analog signal
- Electronic device
- Used to transmit information over telephone line
- Types
  - Internal
  - External

14. Explain Bus topology

- Simplest
- All connected to main cable
- Co-axial cable
- Sent to all nodes, but receives only intended
- **Advantages**
  - Easy to connect
  - Less cable length
  - New node addition is easy
  - Less maintenance cost
- **Disadvantages**
  - If main cable damages, whole network break down
  - Terminators required
  - Speed will be decreased by adding more nodes
  - Multiple sharing not possible
  - Difficult to find problem

15. Explain star topology

- Most common
- All are connected to central node. Central device can be,
  - Hub
  - Switch
  - Router
- UTP or STP cable – Unshielded/Shielded Twisted Pair can be used.
- More secure.
- More costly.
- Better speed than bus topology.
- A failure doesn't affect whole network

16. Explain satellite link

- Uses Microwave signals.
- It works like relay system: Re-transmit the ground signals
- Satellite type is synchronous satellite
- Time saving & Fast implementation
- Broad coverage
- Prone to natural disasters like, flood, earthquake etc.

## 17. Benefits of Microwave network

- Benefits
- Cheaper than optical fiber
- Land acquisition not required
- Communication over difficult terrain
- Over ocean

## 18. Disadvantages of Microwave network

- Interference with out-of-phase signals
- Susceptible to weather
- Bandwidth allocation is limited
- Cost of design, implementation, maintenance is high

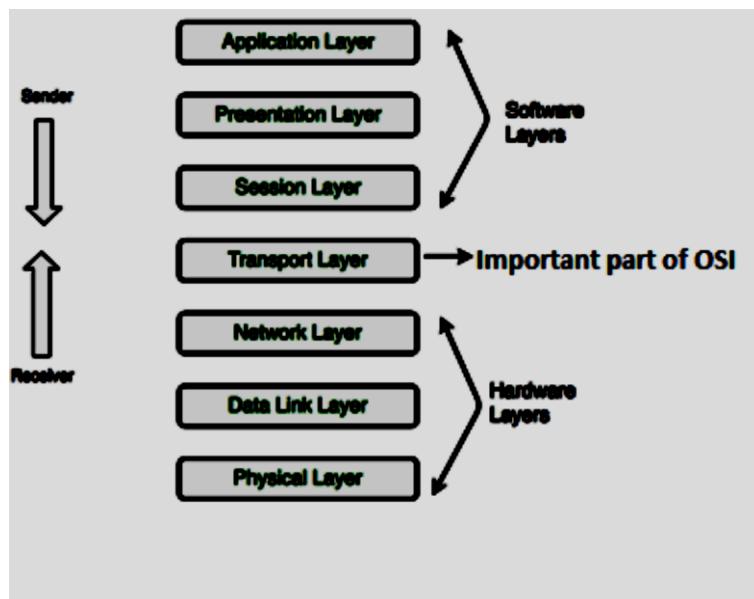
## 19. What is Personal Area Network?

- Personal area network is a very short distance network.
- It can be established between computer handheld devices, Camera, Keyboard, Mouse, Cell phone etc.
- Proximity is up to 10 mtrs
- Connectivity
  - Wired: USB
  - Wireless: Bluetooth, IR
- Used for transferring files, photos, videos etc.

## NETWORK PROTOCOLS

### The OSI Reference Model

The Open Systems Interconnection (OSI) reference model was developed by the International Standards Organisation (ISO) as a model for computer communications architectures, various protocols were standardized by this model. the Communication process is divided under this model within seven layers.



- Physical layer -It is related with physical transmission , hubs , repeaters works at this layer.
- Data link layer -Ensures flow control of bits , error detection , connection between various network topologies devices . Bridges , switches work at this layer , Ethernet is an example of a data link layer protocol.
- Network layer – It decide the routes of packets on internet using IP address . Routers work at this layer.
- Transport layer -It ensures delivery of data with error free and duplicity.
- Session layer –Connection is established by this layer between end system of applications so that communication can be done between them .

- Presentation layer –It resolves data representation related issues of different end systems ,it encodes the data in standard format , using encryption , compression .
- Application layer –The different parts of application specially on distributed systems are managed by this layer.

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

- It is an application level protocol.
- It provide communication between web browser and web server.
- It is client server protocol ie. request is sent by browser and hypermedia documents will be provided by server.
- It can work as generic protocol.
- It is used to deliver data (HTML files, image files, query results, etc.) on the different websites.

### **FTP (File Transfer Protocol)**

- It works on application layer of OSI model.
- It is used to transfer the files on internet.
- Making projects being on long distance is possible with transferring files using this protocol.
- FTP ensures to use different conventions while naming files.
- It provides secure login method for effective transfer of files.
- It provides the different directory structure for different files.

### **Point to Point Protocol (PPP)**

- It works on data link layer of OSI model.
- It establishes direct and dedicate connection between two devices .
- Example – Two routers direct communicate using PPP.
- Our home computer when connects to internet with ISP through modem then there PPP protocol is used.
- For using PPP duplex mode is required for communicating devices , it also ensures data integrity and security , failing which using acknowledge method it resends the lost or damaged data .

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

- SMTP works on application layer of OSI model.
- It is used for email services .
- It is not concerned with message of email , it only uses the information contained in email header .
- Outgoing emails are collected in queue .
- SMTP takes emails one by one from queue and send messages to destination as per address mentioned in header of the mail .
- After sending the mail to destination it removes the messages from outgoing queue , and same time it also removes the receiver's address from mails destination list.
- At receiving end SMTP receiver program sends the mails mail to their corresponding mail boxes .

### **Transmission control protocol / Internet Protocol ( TCP / IP )**

- TCP / IP works with 4 layers i.e. Network , Internet , Transport and Application.
- It uses client – server model of communication .
- TCP breaks the messages in packets .
- IP decides the destination addresses for packets .
- Packets are sent on internet they reach at destination by routing various intermediate nodes and paths .
- Different packets of same message will travel through different paths depending on traffic at particular path.

- TCP is responsible for ordering the packets at receiving and destination.

### **Post office Protocol Version 3 (POP 3)**

- It works on application layer of OSI model .
- It is used to transferring messages from email server to email client .
- It is used to access the remote serve email and receive and filters the email at appropriate user folders.
- Messages are downloaded from mail server to hard disk using this protocol.

### **Hypertext Transfer Protocol Secure (HTTPS)**

Hyper Text Transfer Protocol Secure (HTTPS) is the secure version of HTTP, the protocol over which data is sent between your browser and the website that you are connected to. The 'S' at the end of HTTPS stands for 'Secure'. It means all communications between your browser and the website are encrypted.

### **Remote Access Protocol (Telnet)**

This protocol helps a user (Telnet Client) to log in at a remote computer (Telnet Server) and function as if he/she were connected directly to that computer. Telnet is the main internet protocol for creating a connection with a remote machine. It allows you to connect to remote computers (called remote hosts) over a TCP/IP network (such as the Internet). Once your telnet client establishes a connection to the remote host, your client becomes a virtual terminal, allowing you to communicate with the remote host from your computer with whatever privileges you may have been granted to the specific application and data on that host computer.

### **VoIP**

VOIP: Voice over Internet Protocol (Voice over IP, VoIP and IP telephony) is a group of technologies for the delivery of voice communications and multimedia sessions over Internet. The terms Internet telephony, broadband telephony, and broadband phone service specifically refer to the provisioning of communications services (voice, fax, SMS, voice-messaging) over the public Internet, rather than via the public switched telephone network (PSTN). This method of making phone calls is much cheaper than convectional way because the service of Telecommunication Company is not used.

### **Wireless/Mobile Communication protocol**

**GSM:** (Global System for Mobile communication): is a digital mobile telephony system that is widely used in Europe and other parts of the world. GSM is a second generation (2G) standard for mobile networks. The GSM standard has given birth to wireless services like General Packet Radio Service (GPRS) and Enhanced Data Rates for GSM Evolution (EDGE). GSM uses a variation of time division multiple access (TDMA) and is the most widely used telephony technology.

**GPRS** (General Packet Radio Service): is a packet oriented mobile data service on the 2G and 3G GSM communications. GPRS usage is typically charged based on volume of data transferred.

**WLL (Wireless Local Loop):** Wireless local loop (WLL), is the use of a wireless communications link for delivering telephone service (POTS) or Internet access (broadband) to telecommunications

customers. This system is based on radio networks which provide services like telephone in remote areas. Different types of wireless local loop include Broadband Wireless Access, Radio in the Loop, Fixed Radio Access and Fixed Wireless Access. Microwave frequencies are used to provide wireless broadband connection. The local loop can be referred as "last mile" of the telecom network which resides between Central office (CO) and user's location.

### **Generations of Mobile technology:**

- 1G technology was used in the first mobile phones. 1G used analog radio signals. 1G was introduced in 1980s
- 2G technology was introduced in 1992. 2G technology used a digital format and introduced text messaging. 2G also introduced data services for mobiles, starting with SMS.
- 3G technology has introduced more efficient ways of carrying data, making it possible to have faster web- services, live chat, fast downloading, video conferencing etc. over mobile phones. Today we are living in the world of 3G.
- 4G technology unlike previous generations of mobile technology, 4G mobile technology will be used for internet access on computers also, and it will be totally wireless. 4G will provide internet access, high quality streaming video and "anytime, anywhere" voice and data transmission at a much faster speed than 3G. The "anytime, anywhere" feature of 4G is also referred to as "MAGIC" (Mobile multimedia; Anytime/anywhere; Global mobility support; Integrated wireless solution; Customized personal services).

5G technology - It can provide higher speed, lower latency and greater capacity than 4G LTE networks. It is one of the fastest, most robust technologies the world has ever seen.

In healthcare, 5G technology and Wi-Fi 6 connectivity will enable patients to be monitored via connected devices that constantly deliver data on key health indicators, such as heart rate and blood pressure. I

### **Introduction to Web Services**

**WWW (World Wide Web):** WWW is a network of web servers. It can be defined as a hypertext information retrieval system on the Internet. It is the universe of the information available on the internet. All the web pages on WWW use HTTP for making hypertext jumps.

### **HTML (Hypertext Markup Language)**

- HTML is a markup language to create and design webcontent.
  - It is used to **display the data in formatted manner.**
  - It has a variety of **predefined tags** and attributes for defining the layout and structure of the web document.
  - HTML is **not casesensitive**
  - A HTML document has the extension **.htm or .html.**
  - HTML documents have a **head and body**.
- ```
<html>
<head>.....
</head>
<body>.....
</body>
</html>
```

XML (eXtensible Markup Language)

- XML is a markup language that is self- descriptive.
 - It is used to carry, store or exchange data.
 - It does not have predefined tags. It is possible to define new tags in XML. It allows the programmer to use customized tags.
 - XML is case sensitive.
 - A HTML document has the extension **.xml**
 - XML documents form a tree structure.
- ```
<root>
<child>
<subchild>.....
</subchild>
</child>
</root>
```

**Domain Names:** Every computer on the network has a unique numeric address assigned to it which is a combination of four numbers from 0-255 separated by a dot. For example, 59.177.134.72 since it is practically impossible for a person to remember the IP addresses. A system has been developed which assigns domain names to web servers and maintains a database of these names and corresponding IP addresses on DNS (Domain Name Service) server.

Examples of some domain names are cbse.nic.in, indianrailway.gov.in etc. A domain name usually has more than one part for example, in the domain name www.cbse.nic.in

- in is the primary domain name
- nic is the sub-domain of in
- cbse is the sub-domain of nic
- www indicates the server is on world wide web.

Top level domains are further divided into two categories:

Generic Domain Names:	Country Specific Domain Names:
.com - commercial business	.in - India
.edu - Educational institutions	.au - Australia
.gov - Government agencies	.ca - Canada
.mil - Military	.ch - China
.net - Network organizations	.nz - New Zealand
.org - Organizations (non-profit	.pk - Pakistan
	.jp - Japan
	.us - United States of America

**URL (Uniform resource locator):** A URL is a formatted text string used to identify a network resource on the Internet. Network resources can be plain Web

pages, text documents, graphics, downloadable files, services or programs. Every network resource on the web has a unique URL in the following format:

Protocol: // domain name /path / file name



The URL text string consists of three parts:

- ❖ Network Protocol: The network protocol identifies the protocol to be used to access the network resource. These strings are short names followed by the three characters ': //'. Some examples of protocols include http, gopher, ftp and mailto.
- ❖ Domain name: It identifies the host/server that holds the resource. For example: www. School.com is a domain name.
- ❖ Resource Location: It consists of the path or directory and the file name of resource. For example in the URL : http://www.school.com/syllabus/preprimary/nursery.htm the file nursery.htm is stored in the sub directory preprimary, of the directory syllabus on the server www.school.com

**Website:** Related web pages from a single web domain is termed as a website. A website has multiple web pages providing information about a particular entity.

**Web browser:** Web browser is software program to navigate the web pages on the internet. A browser interprets the coding language of the web page and displays it in graphic form. Internet works on client -server model. A web browser is a client which requests the information from the web server. The web server sends the information back to the client. Some of the web browsers are: Netscape Navigator, Internet Explorer, Mozilla Firefox etc.

**Web Server:** A Web server is a computer or a group of computers that stores web pages on the internet. It works on client/server model. It delivers the requested web page to web browser. Web servers use special programs such as Apache or IIS to deliver web pages over the http protocol.

Each server has a unique IP address and domain name. In order to access a webpage, the user writes the URL of the site on the address bar of the browser. The machine on which the browser is running sends a request to the IP address of the machine running the web server for that page. Once the web server receives that request, it sends the page content back to the IP address of the computer asking for it. The web browser then translates that content into all of the text, pictures, links, videos, etc. A single web server may support multiple websites or a single website may be hosted on several linked servers.

**Web hosting:** Web hosting is the process of uploading/saving the web content on a web server to make it available on WWW. In case an individual or a company wants to make its website available on the internet, it should be hosted on a web server.

**Web page:** Web page is an electronic document designed using HTML. It displays information in textual or graphical form. It may also contain downloadable data files, audio files or video files. Traversal from one webpage to another web page is possible through hyperlinks. A web page can be classified into two types:

<p>Static web page: A web page which displays same kind of information whenever a user visits it is known as a static web page. A static web page generally has <b>.htm or .html</b> as extension</p>	<p>Dynamic web page: An interactive web page is a dynamic webpage. A dynamic web page uses scripting languages to display changing content on the web page. Such a page generally has <b>.php, .asp or .jsp</b> as extension.</p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## **SUMMARY**

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

- It is used to deliver data (HTML files, image files, query results, etc.) on the different websites.

### **FTP (File Transfer Protocol)**

- It is used to transfer the files on internet.

### **Point to Point Protocol (PPP)**

- Our home computer when connects to internet with ISP through modem then there PPP protocol is used.

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

- SMTP takes emails one by one from queue and send messages to destination as per address mentioned in header of the mail .

### **Transmission control protocol / Internet Protocol ( TCP / IP )**

- TCP breaks the messages in packets .
- IP decides the destination addresses for packets .

### **Post office Protocol Version 3 (POP 3)**

- It is used to transferring messages from email server to email client .

### **Hypertext Transfer Protocol Secure (HTTPS)**

- The 'S' at the end of HTTPS stands for 'Secure'. It means all communications between your browser and the website are encrypted.

### **Remote Access Protocol (Telnet) :**

- This protocol helps a user (Telnet Client) to log in at a remote computer (Telnet Server) and function as if he/she were connected directly to that computer.

### **VOIP: Voice over Internet Protocol**

- This is a group of technologies for the delivery of voice communications and multimedia sessions over Internet.

### **GSM: (Global System for Mobile communication)**

- GSM) is a second generation (2G) standard for mobile networks. The GSM standard has given birth to wireless services like General Packet Radio Service (GPRS) and Enhanced Data Rates for GSM Evolution (EDGE).

### **GPRS (General Packet Radio Service)**

- This is a packet oriented mobile data service on the 2G and 3G GSM communications. GPRS usage is typically charged based on volume of data transferred.

### **WLL (Wireless Local Loop)**

Wireless local loop (WLL), is the use of a wireless communications link for delivering telephone service (POTS) or Internet access (broadband) to telecommunications customers.

### **Mobile Technologies**

- 1G technology was used in the first mobile phones. 1G used analog radio signals. 1G was introduced in 1980s
- 2G technology 2G also introduced data services for mobiles, starting with SMS.
- 3G technology has introduced more efficient ways of carrying data, making it

possible to have faster web- services, live chat, fast downloading, video conferencing etc. over mobile phones. Today we are living in the world of 3G.

➤ 4G technology - 4G provides internet access, high quality streaming video and "anytime, anywhere" voice and data transmission at a much faster speed than 3G.

➤ 5G technology - It can provide higher speed, lower latency and greater capacity than 4G LTE networks. It is one of the fastest, most robust technologies the world has ever seen.

**WWW (World Wide Web):** WWW is a network of web servers.

**HTML (Hypertext Markup Language)**

- HTML is a markup language to create and design web content.
- It is used to display the data in formatted manner.

**XML (eXtensible Markup Language)**

- XML is a markup language that is self- descriptive.
- It is used to carry, store or exchange data.

**Domain Names:** A system has been developed which assigns domain names to web servers and maintains a database of these names and corresponding IP addresses on DNS (Domain Name Service) server.

**URL- Uniform resource locator**

**Examples – Website :[www.rajkot.kvs.ac.in](http://www.rajkot.kvs.ac.in)**

- **Web server** :Apache HTTP Server , Microsoft Internet Information Services (IIS) , Sun Java System Web Server.

Web Browser : Google Chrome , Mozilla Firefox , Internet Explore .

Definition :

Web Page : page of website .

Web Hosting :Web site hosting / Web hosting / and Webhosting is **the business of housing, serving, and maintaining files for one or more Web sites.**

### **Question Bank** **1 Mark Questions**

Q1 Mr. Ramesh is not able to identify the Domain Name in the given URL. Identify and write it for him.

Q2 Expand following –

- (i) HTTPS
- (ii) TDMA

Q3 Name any two most popularly used search engines.

Q4 Given below two addresses

- (i) 59.177.134.72
- (ii) <http://www.ccc.nic.in/welcome.htm>

Identify which one of the above is an IP address and which one is a URL?

Q5 Mr. Mahesh wants to prevent unauthorised access to/from his company's local area network. Write the name of system (software/hardware), which he should install.

Q6. Write any one disadvantages of VoIP.

Q7 Write the full form of following –

- 1. GSM
- 2. GPRS

Q8. What is Error-404?

Q9. Name an open source web browser.

Q 10. Name any two common web browser.

Q 11 What is a protocol? Which protocol is used to search information from Internet using an Internet browser?

## **2 Mark Questions**

Q1 Expand following –

- (i) XML
- (ii) GSM

Q2 What is an IP address , explain with example ?

Q3 Write a short note on WWW.

Q4 Ramesh is a class 12 computer science student.

He is confused between 'web' and 'internet'. As a friend of Ramesh, explain her the difference between web and internet.

Q5 Write short note on VoIP.

Q6 Write two advantages of VoIP.

Q7. Write a short note on web hosting.

Q8 Define browser.

Q9. What is DNS?

Q10 What is work of SMTP explain.

Q 11 What is work of TCP / IP explain .

Q12 What is Telnet expain . ?

Q 13 write full form of following –

- (i) PPP
- (ii) FTP
- (iii) POP
- (iv) WLL

## **3 Mark Questions**

Q 1 Identify the domain name, protocol, sub-domain from the following domain.

**<https://www.csiplearninghub.com/class-11-ch-1/mcq>**

Q 2 . Suresh is running a business. He has studied till class 10. He wants to launch a website for his business. He is confused between static and dynamic web page. As a website developer, help him to understand the difference between static and dynamic web page.

Q 3. Define the following terms :

1. Website
2. Webpage
3. Web server

Q4. Write the steps to host a website.

Q5 – Write note on 1G , 2G , 3G of mobile technologies.

Q6 – Write note on 4G , 5G of mobile technologies.

Q7 Write differences between HTML and XML ?

Q8 Explain working of SMTP ?

Q9 Explain working of TCP / IP ?

Q10 What do you mean by domain name explain with example .

Q 11 Explain PPP ( Point to Point) Protocol .

## CASE STUDY BASED QUESTIONS

**Case 1:** Web server is a special computer system running on HTTP through web pages. The web page is a medium to carry data from one computer system to another. The working of the webserver starts from the client or user. The client sends their request through the web browser to the webserver. Web server takes this request, processes it and then sends back processed data to the client. The server gathers all of our web page information and sends it to the user, which we see on our computer

system in the form of a web page. When the client sends a request for processing to the web server, a domain name and IP address are important to the webserver. The domain name and IP address are used to identify the user on a large network.

1.

1. Web servers are:
  1. IP addresses
  2. Computer systems
  3. Webpages of a site
  4. A medium to carry data from one computer to another
2. What does the webserver need to send back information to the user?
  1. Home address
  2. Domain name
  3. IP address
  4. Both b and c
3. What is the full form of HTTP?
  1. Hypertext Transfer Protocol
  2. Hypertext Transfer Procedure
  3. Hyperlink Transfer Protocol
  4. Hyperlink Transfer Procedure
4. The \_\_\_\_\_ translates internet domain and host names to IP address
  1. Domain name system
  2. Routing information protocol
  3. Google
  4. Network time protocol
5. Computer that requests the resources or data from other computer is called as \_\_\_\_\_ computer
  1. Server
  2. Client
  3. None of the above
  4. a and b
6. DNS stands for:
  1. Domain Name Security
  2. Domain Number System
  3. Document Name System
  4. Domain Name System

**Case 2:** In 1989, Tim Berners Lee, a researcher, proposed the idea of World Wide Web). Tim Berners Lee and his team are credited with inventing Hyper Text Transfer Protocol (HTTP), HTML and the technology for a web server and a web browser. Using hyperlinks embedded in hypertext the web developers were able to connect web pages. They could design attractive webpages containing text, sound and graphics. This change witnessed a massive expansion of the Internet in the 1990s.

1. What is a web browser?
  1. A program that can display a webpage
  2. A program used to view HTML documents
  3. It enables a user to access the resources of internet
  4. All of the above
2. Dynamic web page \_\_\_\_\_
  - a) is same every time whenever it displays
  - b) generates on demand by a program or a request from browser
  - c) both is same every time whenever it displays and generates on demand by a

program or a request from browser  
d) is different always in a predefined order

3. URL stands for \_\_\_\_\_
- a) unique reference label
  - b) uniform reference label
  - c) uniform resource locator
  - d) unique resource locator

4. A web cookie is a small piece of data that is \_\_\_\_\_  
 a) sent from a website and stored in user's web browser while a user is browsing a website  
 b) sent from user and stored in the server while a user is browsing a website  
 c) sent from root server to all servers  
 d)1.
5. Google Chrome is an good –  
 1. Web server.  
 2. Web browser.  
 3. Web protocol.  
 4. Subset of website sent from the root server to other root servers

### **ANSWER KEY** **1 Mark Questions**

Q1 Mr. Ramesh is not able to identify the Domain Name in the given URL. Identify and write it for him.

Ans1 Domain in <http://www.cbsenec.in/aboutus.htm> is "www.cbsenec.in".

Q2 Expand following –

- (iii) HTTPS
- (iv) TDMA

Ans2

- (i) Hypertext Transfer Protocol Secure (HTTPS)
- (ii) Time division multiple access (TDMA)

Q3 Name any two most popularly used search engines.

Ans3 Two most popular search engines are google and grotel.

(url:<http://www.google.com> and <http://grotel.com>)

Q4 Given below two addresses

- (i) 59.177.134.72
- (ii) <http://www.ccc.nic.in/welcome.htm>

Identify which one of the above is an IP address and which one is a URL?

Ans4 i) IP Address 59.177.134.72

(ii) URL <http://www.ccc.nic.in/welcome.htm>

Q5 Mr. Mahesh wants to prevent unauthorised access to/from his company's local area network. Write the name of system (software/hardware), which he should install.

Ans 5 He should install firewall system to prevent unauthorised access.

Q6. Write any one disadvantages of VoIP.

**Ans6. The disadvantage of VoIP is that its call quality is dependent on Internet connection speed. Slow Internet connection will lead to poor quality voice calls.**

Q7 Write the full form of following –

- 2. GSM
- 3. GPRS

Ans 7.

- 1. Global System for Mobile communication
- 2. General Packet RadioServices
- 3.

Q8. What is Error-404?

**Ans8. If the server is not able to locate the page, it sends a page containing the error message (Error 404 – page not found) to the client's browser.**

Q9. Name an open source web browser.

**Ans 9. Mozilla Firefox is an open source web browser.**

Q 10. Name any two common web browser.

**Ans 10 . Some of the commonly used web browsers are Google Chrome, Internet Explorer, Mozilla Firefox, Opera, etc.**

Q 11 What is a protocol? Which protocol is used to search information from Internet using an Internet browser? .

Ans 11 . Protocol refers to the rules which are applicable for a network. These protocols define the standardised format for data packets to be transmitted over the network, techniques for detecting errors and correcting them etc.

The protocol used to search the information from Internet using browser is HyperText Transfer Protocol (HTTP).

## **2 Mark Questions**

Q1 Expand following –

- (iii) XML
- (iv) GSM

Ans1 Answer:

- (i) Xtensible Markup Language.
- (ii) Global System for Mobile Communication.

Q2 What is an IP address?

Ans2. An Internet Protocol address (IP address) is a numerical unique address of a device in a network. IP is a datagram-oriented connectionless protocol, therefore each packet must contain a header with the source IP address, the destination IP address, and other data in order to be delivered successfully.

Q3 Write a short note on WWW.

**Ans3. The World Wide Web (WWW) or web in short, is an ocean of information, stored in the form of trillions of interlinked web pages and web resources. The resources on the web can be shared or accessed through the Internet.**

Q4 Ramesh is a class 12 computer science student.

He is confused between 'web' and 'internet'. As a friend of Ramesh, explain her the difference between web and internet.

**Ans4. The Internet as we know is the huge global network of interconnected computers, which may or may not have any file or webpage to share with the world. The web on the other hand is the interlinking of a collection of WebPages on these computers which are accessible over the Internet.**

Q5 Write short note on VoIP.

**Ans5. Voice over Internet Protocol or VoIP, allows us to have voice call (telephone service) over the Internet, i.e., the voice transmission over a computer network rather than through the regular telephone network. It is also known as Internet Telephony or Broadband Telephony.**

Q6 Write two advantages of VoIP.

**Ans6. Advantages of VoIP are :**

1. These services are either free or very economical, so people use them to save on cost. That is why these days even international calls are being made using VoIP.
2. VoIP call(s) can be received and made using IP phones from any place having Internet access.

Q7. Write a short note on web hosting.

**Ans 7. Web hosting is a service that allows us to put a website or a web page onto the Internet, and make it a part of the World Wide Web. Once a website is created, we need to connect it to the Internet so that users across the globe can access.**

Q8 Define browser.

**Ans 8. A browser is a software application that helps us to view the web page(s). In other words, it helps us to view the data or information that is retrieved from various web servers on the Internet.**

Q9. What is DNS?

**Ans 9. The domain name system (DNS) is a service that does the mapping between domain name and IP address. When the address of a website is entered in a browser, the DNS finds out the IP address of the server corresponding to the requested domain name and sends the request to that server.**

Q10 What is work of SMTP explain.

Ans 10. SMTP or Simple Mail Transfer Protocol is an application that is used to send, receive, and relay outgoing emails between senders and receivers. When an email is sent, it's transferred over the internet from one server to another using SMTP. In simple terms, an SMTP email is just an email sent using the SMTP server.

Q 11 What is work of TCP / IP explain .

Ans 11 . TCP breaks the message into packets and IP decides the destination address for packets .

Q12 What is Telnet expain . ?

Ans 12 Telnet is a network protocol used to virtually access a computer and to provide a two-way, collaborative and text-based communication channel between two machines. It follows a user command Transmission Control Protocol/Internet Protocol (TCP/IP) networking protocol for creating remote sessions.

Q 13 write full form of following –

- (v) PPP
- (vi) FTP
- (vii) POP
- (viii) WLL

Ans 13 (i) PPP – Point to Point Protocol.

(ii) FTP – File Transfer Protocol.

(iii) POP – Post office protocol.

WLL – Wireless in Local loop .

### 3 Mark Questions

Q 1 Identify the domain name, protocol, sub-domain from the following domain.

**https://www.csiplearninghub.com/class-11-ch-1/mcq**

**Ans 1 . Domain name = www.csiplearninghub.com**

**Sub Domain Name = www**

**Protocol = https**

**Q 2 . Suresh is running a business. He has studied till class 10. He wants to launch a website for his business. He is confused between static and dynamic web page. As a website developer, help him to understand the difference between static and dynamic web page.**

**Ans 2. A static webpage is one whose content always remains static, i.e., does not change for person to person. A dynamic web page is one in which the content of the web page can be different for different users. They are more complex and thus take more time to load than static web pages.**

**Q 3. Define the following terms :**

4. Website
5. Webpage
6. Web server

**Ans 3. Website : A website (usually referred to as a site in short) is a collection of web pages related through hyperlinks, and saved on a web server. A visitor navigates from one page to another by clicking on hyperlinks.**

**Webpage : A web page (also referred to as a page) is a document on the WWW that is viewed in a web browser. Basic structure of a web page is created using HTML (Hyper Text Markup Language) and CSS.**

**Web Server: A web server is used to store and deliver the contents of a website to clients such as a browser that request it. A web server can be software or hardware.**

**Q4. Write the steps to host a website.**

**Ans 4. Steps to host a website are :**

- 1. Select the web hosting service provider that will provide the web server space.**
- 2. Identify a domain name, which best suits our requirement.**
- 3. Create logins with appropriate rights and note down IP address to manage web space.**
- 4. Upload the files in properly organised folders on the allocated space.**
- 5. Get domain name mapped to the IP address of the web server.**

**Q5 – Write note on 1G , 2G , 3G of mobile technologies.**

**Ans 5 -**

- 1G technology was used in the first mobile phones. 1G used analog radio signals. 1G was introduced in 1980s
- 2G technology was introduced in 1992. 2G technology used a digital format and introduced text messaging. 2G also introduced data services for mobiles, starting with SMS.
- 3G technology has introduced more efficient ways of carrying data, making it possible to have faster web- services, live chat, fast downloading, video conferencing etc. over mobile phones. Today we are living in the world of 3G.

**Q6 – Write note on 4G , 5G of mobile technologies.**

**Ans6 : 4G technology unlike previous generations of mobile technology, 4G mobile technology will be used for internet access on computers also, and it will be totally wireless. 4G will provide internet access, high quality streaming video and "anytime, anywhere" voice and data transmission at a much faster speed than 3G. The "anytime, anywhere" feature of 4G is also referred to as "MAGIC" (Mobile multimedia; Anytime/anywhere; Global mobility support; Integrated wireless solution; Customized personal services).**

5G technology - It can provide higher speed, lower latency and greater capacity than 4G LTE networks. It is one of the fastest, most robust technologies the world has ever seen.

In healthcare, 5G technology and Wi-Fi 6 connectivity will enable patients to be monitored via connected devices that constantly deliver data on key health indicators, such as heart rate and blood pressure.

Q7 Write differences between HTML and XML ?

Ans 7.

<b>HTML (Hypertext Markup Language)</b>	<b>XML (eXtensible Markup Language)</b>
<ul style="list-style-type: none"><li>➤ HTML is a markup language to create and design webcontent.</li><li>➤ It is used to <b>display the data in formatted manner</b>.</li><li>➤ It has a variety of <b>predefined tags</b> and attributes for defining the layout and structure of the web document.</li><li>➤ HTML is <b>not casesensitive</b></li><li>➤ A HTML document has the extension <b>.htm or .html</b>.</li><li>➤ HTML documents have a <b>head and body</b>. <code>&lt;html&gt;   &lt;head&gt;.....   &lt;/head&gt;   &lt;body&gt;.....   &lt;/body&gt; &lt;/html&gt;</code></li></ul>	<ul style="list-style-type: none"><li>➤ XML is a markup language that is self-descriptive.</li><li>➤ It is used to <b>carry, store or exchange data</b>.</li><li>➤ It does not have predefined tags. It is possible to define new tags in XML. It allows the programmer to use <b>customized tags</b>.</li><li>➤ XML is <b>casesensitive</b>.</li><li>➤ A HTML document has the extension <b>.xml</b></li><li>➤ XML documents form a <b>tree structure</b>. <code>&lt;root&gt;   &lt;child&gt;     &lt;subchild&gt;....   &lt;/subchild&gt;   &lt;/child&gt; &lt;/root&gt;</code></li></ul>

Q8 Explain working of SMTP ?

Ans 8.Simple Mail Transfer Protocol (SMTP)

- SMTP works on application layer of OSI model.
- It is used for email services .
- It is not concerned with message of email , it only uses the information contained in email header .
- Outgoing emails are collected in queue .
- SMTP takes emails one by one from queue and send messages to destination as per address mentioned in header of the mail .
- After sending the mail to destination it removes the messages from outgoing queue , and same time it also removes the receiver's address from mails destination list.
- At receiving end SMTP receiver program sends the mails mail to their corresponding mail boxes .

Q9 Explain working of TCP / IP ?

Ans 9      Transmission control protocol / Internet Protocol ( TCP / IP )

- TCP / IP works with 4 layers ie .Network , Internet , Transport and Application .
- It uses client – server model of communication .
- TCP breaks the messages in packets .
- IP decides the destination addresses for packets .
- Packets are sent on internet they reach at destination by routing various intermediate nodes and paths .
- Different packets of same message will travel through different paths depending on traffic at particular path.
- TCP is responsible for ordering the packets at receiving and destination.

**Q10** What do you mean by domain name explain with example .

**Ans10 Domain Names:** Every computer on the network has a unique numeric address assigned to it which is a combination of four numbers from 0-255 separated by a dot. For example, 59.177.134.72 since it is practically impossible for a person to remember the IP addresses. A system has been developed which assigns domain names to web servers and maintains a database of these names and corresponding IP addresses on DNS (*Domain Name Service*) server.

Examples of some domain names are cbse.nic.in, indianrailway.gov.in etc. A domain name usually has more than one part for example, in the domain name www.cbse.nic.in

1. in is the primary domainname
2. nic is the sub-domain ofin
3. cbse is the sub-domain ofnic
4. www indicates the server is on world wideweb

**Q 11** Explain PPP ( Point to Point) Protocol .

**Ans 11**

Point to Point Protocol (PPP)

- It works on data link layer of OSI model.
- It establishes direct and dedicate connection between two devices .
- Example – Two routers direct communicate using PPP.
- Our home computer when connects to internet with ISP through modem then there PPP protocol is used.
- For using PPP duplex mode is required for communicating devices , it also ensures data integrity and security , failing which using acknowledge method it resends the lost or damaged data .

## CASE STUDY BASED QUESTIONS

### **Case 1**

Web servers are:

Ans. 2

What does the webserver need to send back information to the user?

Ans. 4

What is the full form of HTTP?

Ans. 1

The \_\_\_\_\_ translates internet domain and host names to IP address

Ans. 1

Computer that requests the resources or data from other computer is called as \_\_\_\_\_ computer

ans. 2

DNS stands for:

Ans. 4

### **Case 2 1.** What is a web browser?

Ans. 4

2. Dynamic web page \_\_\_\_\_

ans. 2

3. URL stands for \_\_\_\_\_

ans. 3

4. A web cookie is a small piece of data that is \_\_\_\_\_

ans. 1

5. Google Chrome is an good –  
Ans : Web Browser .

## Unit III - DBMS concepts and SQL

### Database

A database is an organized collection of structured data, stored electronically in a computer system.

### Relational Database

A relational database is a collection of data items organized as logically related tables.

The diagram illustrates a relational database structure. At the top, a box contains the heading "Table DEPT". Below it is a table with three columns: DEPTNO, DNAME, and LOC. The data is as follows:

DEPTNO	DNAME	LOC
20	RESEARCH	DALLAS
30	SALES	CHICAGO

An arrow points from the bottom right corner of the DEPT table towards the top left corner of the EMP table, which is located below it. The EMP table has four columns: EMPNO, ENAME, JOB, and DEPTNO. The data is as follows:

EMPNO	ENAME	JOB	DEPTNO
7329	SMITH	CEO	20
7499	ALLEN	VP_SALES	30
7521	WARD	MANAGER	30
7566	JONES	SALESMAN	30
7691	OSTER	SALESMAN	20

### Database Management System

The software required to manage a database is known as a database management system (DBMS).

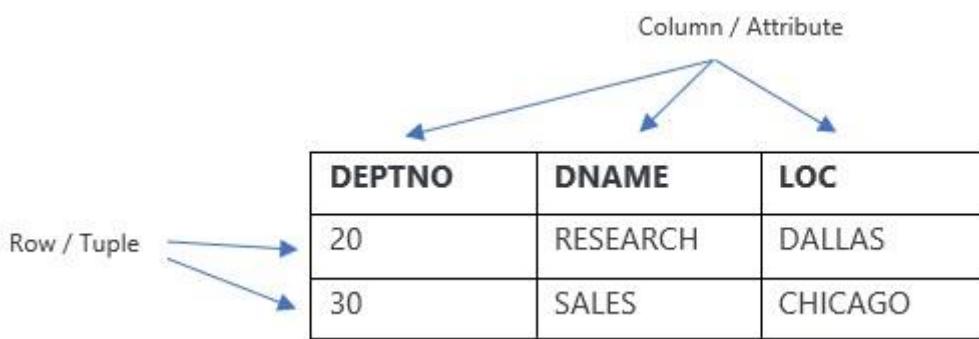
A DBMS serves as an interface between the database and its end users, allowing users to retrieve, update, and manage how the information is organized and optimized.

A DBMS also facilitates oversight and control of databases, enabling a variety of administrative operations such as performance monitoring, tuning, and backup and recovery.

handle/manipulate these table/relations is known as Relational Database Management System (RDBMS). Some examples of popular DBMSs are MySQL, Microsoft Access, Microsoft SQL Server, FileMaker Pro, Oracle Database, and dBASE

### Table/Relation

A group of rows and columns form a table. The horizontal subset of the Table is known as a Row/Tuple. The vertical subset of the Table is known as a Column/an Attribute.



A relation in a database has the following characteristics:

- Values are atomic.
- Column values are of the same kind.
- Each row is unique.
- The sequence of columns is insignificant.
- The sequence of rows is insignificant.
- Each column must have a unique name.

### Database Terminology

Degree

No. of columns of Table.

### **Cardinality**

No. of Rows of Table

### **Domain**

A domain is the collection of values that a data element may contain.

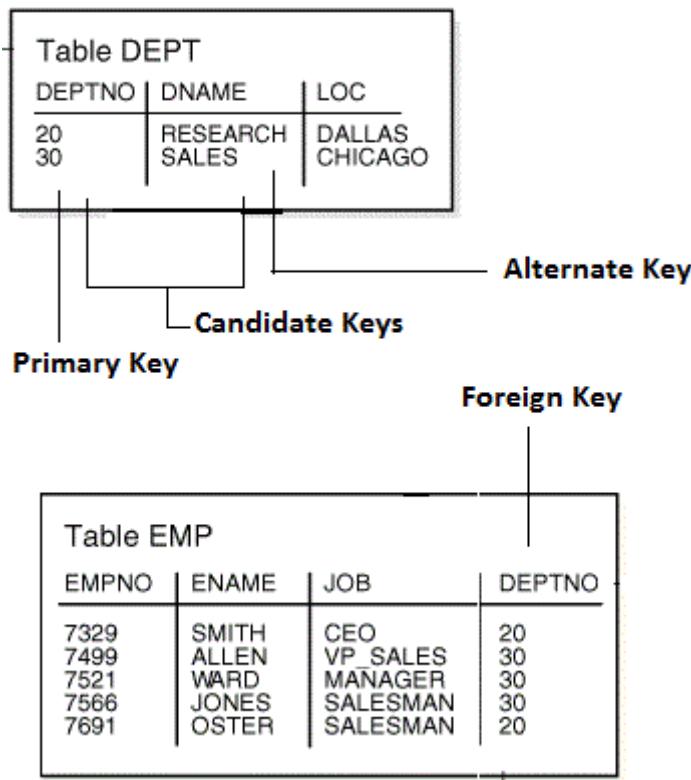
### **Key**

An Attribute/group of attributes in a table that identifies a tuple uniquely is known as a key.

A table may have more than one such attribute/group of identifies that identifies a tuple uniquely, all such attributes(s) are known as **Candidate Keys**.

Out of Candidate keys, one is selected as **Primary key**, and others become **Alternate Keys**.

A **Foreign Key** is defined in a second table, but it refers to the primary key in the first table.



## SQL - Structured Query Language

Structured Query Language (SQL) is a specialized language for accessing and manipulating databases.

SQL commands are classified by function:

**Data definition language (DDL)** - used to define or change database structure(s) (e.g., CREATE, ALTER, DROP)

**Data manipulation language (DML)** - used to select or change data (e.g., INSERT, UPDATE, DELETE, SELECT)

**Transaction Control language** - used to control logical units of work (e.g., COMMIT, ROLLBACK)

**CREATE DATABASE statement:** The CREATE DATABASE statement is used to create a new SQL database. Syntax:

```
CREATE DATABASE databasename;
CREATE DATABASE school;
```

**SHOW DATABASES statement**

The SHOW DATABASES statement is used to know the names of existing databases.  
SHOW DATABASES;

**USE statement**

In order to use the database, the following SQL statement is required. Syntax:  
USE databasename;

**DROP DATABASE statement**

The DROP DATABASE statement is used to delete a database from system. Syntax:  
DROP DATABASE databasename;  
DROP DATABASE school;

## **CREATE TABLE statement**

A database consists of many tables. In order to create a table in database

CREATE TABLE statement is used. Syntax:

```
CREATE TABLE table_name
(
column_name1 data_type (size) constraint,
column_name2 data_type (size) constraint,
column_name3 data_type (size) constraint,
....;
);
```

### **Data Types of attribute (column)**

char(n)	A FIXED length string. The n specifies the column length. The parameter n can be from 0 to 255. Default is 1
varchar(n)	A VARIABLE length string. The n parameter specifies the maximum column length in characters - can be from 0 to 65535
int	An integer. Range is from -2147483648 to 2147483647.
float	A floating point number.
date	A date. Format: YYYY-MM-DD.

### Constraints

Constraints are the certain types of restrictions on the data values that an attribute can have.

Constraint	Description
NOT NULL	Ensures that a column cannot have NULL a value
UNIQUE	Ensures that all the values in a column are different
DEFAULT	Sets a default value for a column if no value is specified
PRIMARY KEY	The column which can uniquely identify each row/record in a table.
FOREIGN KEY	The column which refers to value of an attribute defined as primary key in another table

Example :

```
CREATE TABLE student
(
rollnumber int NOT NULL,
name char(25) NOT NULL,
fees int DEFAULT 7000,
dob date,
class char(3),
PRIMARY KEY (rollnumber)
);
```

## **DESCRIBE statement**

Provides a description of the specified table. Syntax:

```
DESCRIBE table_name;
```

```
DESCRIBE student;
```

## **INSERT INTO statement**

Inserting a new row at the bottom of the table.

Syntax:INSERT INTO table\_name  
VALUES (value1, value2, value3,...);

INSERT INTO table\_name (column1, column2, column3,...)  
VALUES(value1,value2,value3,...);

INSERT INTO student VALUES(10, 'Alex', 7800, '1998-10-03','K12');

INSERT INTO student(rollnumber, name, fees, dob, class) values(11, 'Peter', 6700,  
'1997-11-15',

### **QUESTION NO 1 TO 10 OF ONE MARKS.**

1. What is the full form of DBMS?

- a) Data of Binary Management System
- b) Database Management System
- c) Database Management Service
- d) Data Backup Management System

2. What is a database?

- a) Organized collection of information that cannot be accessed, updated, and managed
- b) Collection of data or information without organizing
- c) Organized collection of data or information that can be accessed, updated, and managed
- d) Organized collection of data that cannot be updated

3. What is DBMS?

- a) DBMS is a collection of queries
- b) DBMS is a high-level language
- c) DBMS is a programming language
- d) DBMS stores, modifies and retrieves data

4. Who created the first DBMS?

- a) Edgar Frank Codd
- b) Charles Bachman
- c) Charles Babbage
- d) Sharon B. Codd

5. Which type of data can be stored in the database?

- a) Image oriented data
- b) Text, files containing data
- c) Data in the form of audio or video
- d) All of the above

6. In which of the following formats data is stored in the database management system?

- a) Image
- b) Text
- c) Table
- d) Graph

7. What does an RDBMS consist of?

- a) Collection of Records
- b) Collection of Keys

- c) Collection of Tables  
d) Collection of Fields

8. The values appearing in given attributes of any tuple in the referencing relation must likewise occur in specified attributes of at least one tuple in the referenced relation, according to \_\_\_\_\_ integrity constraint.

- a) Referential  
b) Primary  
c) Referencing  
d) Specific

9. The DBMS acts as an interface between \_\_\_\_\_ and \_\_\_\_\_ of an enterprise-class system.

- a) Data and the DBMS  
b) Application and SQL  
c) Database application and the database  
d) The user and the software

10. The ability to query data, as well as insert, delete, and alter tuples, is offered by

- a) TCL (Transaction Control Language)  
b) DCL (Data Control Language)  
c) DDL (Data Definition Langauge)  
d) DML (Data Manipulation Langauge)

### **QUESTION NO 11 TO 20 OF (TWO MARKS)**

11. Differentiate between char(n) and varchar(n) data types with respect to databases.

12. Consider the table, MOVIEDETAILS given below:

Table: MOVIEDETAILS

MOVIEID	TITLE	LANGUAGE	RATING	PLATFORM
M001	Minari	Korean	5	Netflix
M004	MGR Magan	Tamil	4	Hotstar
M010	Kaagaz	Hindi	3	Zee5
M011	Harry Potter and the Chamber of Secrets	English	4	Prime Video
M015	Uri	Hindi	5	Zee5
M020	Avengers: Endgame	English	4	Hotstar

- a. Identify the degree and cardinality of the table.  
b. Which field should be made the primary key? Justify your answer.

13. Charu has to create a database named MYEARTH in MYSQL.

She now needs to create a table named CITY in the database to store the records of various cities across the globe. The table CITY has the following structure:

Table: CITY

FIELD NAME	DATA TYPE	REMARKS
CITYCODE	CHAR(5)	Primary Key
CITYNAME	CHAR(30)	
SIZE	INTEGER(3)	
AVGTEMP	INTEGER	
POLLUTIONRATE	INTEGER	

POPULATION	INTEGER	
------------	---------	--

14. What is different between WHERE and HAVING clause.  
 15. What is the importance of primary key in a table? Explain with suitable example.  
 16. Differentiate between Primary key and Candidate key.  
 17. Differentiate between Degree and Cardinality  
 18. What are DDL AND DML.  
 19. What is relation. what is difference between a tuple and attribute?  
 20. Differentiate between SQL commands DROP TABLE and DROP VIEW.

### QUESTION NO 21 TO 30 OF (THREE MARKS)

21. A table, ITEM has been created in a database with the following fields:

ITEMCODE, ITEMNAME, QTY, PRICE

Give the SQL command to add a new field, DISCOUNT (of type Integer) to the ITEM table.

Categorize following commands into DDL and DML commands?

INSERT INTO, DROP TABLE, ALTER TABLE, UPDATE...SET

22. Write a output for SQL queries (i) to (iii), which are based on the table: STUDENT given below:

Table: STUDENT

RollNo	Name	Class	DOB	Gender	City	Marks
1	Nanda	X	06-06-1995	M	Agra	551
2	Saurabh	XII	07-05-1993	M	Mumbai	462
3	Sanal	XI	06-05-1994	F	Delhi	400
4	Trisla	XII	08-08-1995	F	Mumbai	450
5	Store	XII	08-10-1995	M	Delhi	369
6	Marisla	XI	12-12-1994	F	Dubai	250
7	Neha	X	08-12-1995	F	Moscow	377
8	Nishant	X	12-06-1995	M	Moscow	489

(i) SELECT COUNT(\*), City FROM STUDENT GROUP BY CITY HAVING COUNT(\*)>1;

(ii) SELECT MAX(DOB),MIN(DOB) FROM STUDENT;

(iii)SELECT NAME,GENDER FROM STUDENT WHERE CITY="Delhi";

23. Write the output of the queries (a) to (c) based on the table, Furniture given below:

Table: FURNITURE

FID	NAME	DATE OF PURCHASE	COST	DISCOUNT
B001	Double Bed	03-Jan-2018	45000	10
T010	Dining Table	10-Mar-2020	51000	5
B004	Single Bed	19-Jul-2021	22000	0
C003	Long Back Chair 6	30-Dec-2016	12000	3
T006	Console Table	17-Nov-2019	15000	12
B006	Bunk Bed	01-Jan-2021	28000	14

1. SELECT SUM(DISCOUNT) FROM FURNITURE WHERE COST>15000
2. SELECT MAX(DATEOFPURCHASE) FROM FURNITURE
3. SELECT \* FROM FURNITURE WHERE DISCOUNT>5 AND FID LIKE "T%";

24. Write queries (a) to (d) based on the tables EMPLOYEE and DEPARTMENT given below:

Table: EMPLOYEE

EMPID	NAME	DOB	DEPTID	DESIG	SALARY
120	Alisha	23-Jan-1978	D001	Manager	75000
123	Nitin	10-Oct-1977	D002	AO	59000
129	Navjot	12-Jul-1971	D003	Supervisor	40000
130	Jimmy	30-Dec-1980	D004	Sales Rep	
131	Faiz	06-Apr-1984	D001	Dep Manager	65000

Table: DEPARTMENT

DEPTID	DEPTNAME	FLOORNO
D001	Personal	4
D002	Admin	10
D003	Production	1
D004	Sales	3

- a. To display the average salary of all employees, department wise.
- b. To display name and respective department name of each employee whose salary is more than 50000.
- c. To display the names of employees whose salary is not known, in alphabetical order.

25. Give a suitable Example of a table with sample data and illustrate primary key and Candidate keys in it.

26. Give a suitable Example of a table with sample data and illustrate primary key and Alternate keys in it.

27. What do you understand by primary key? Give a suitable of Primary Key from a table containing some meaningful data.

28. Differentiate between Candidate Key and Alternate Key in context of RDBMS.

29. Write SQL queries for (i) to (iii) which are based on the table.

TABLE: ACCOUNT

ANO	ANAME	ADDRESS
101	Nirja Singh	Bangalore
102	Rohan Gupta	Chennai
103	Ali Reza	Hyderabad
104	Rishabh Jain	Chennai
105	Simran Kaur	Chandigarh

1. Select ANAME, ADDRESS FROM ACCOUNT;
2. Select ANAME WHERE ANO='101' FROM ACCOUNT;
3. Select DISTINCT ADDRESS FROM ACCOUNT;

30. Write SQL queries for (i) to (iii) which are based on the table.

TABLE: SALES

S NO	SNAME	ADDRESS
501	Ajay Singh	Bangalore
502	Ram Gupta	Chennai
703	Ali	Hyderabad
704	Rishabh	Chennai

1. Alter table sales

Add primary key sales(S No);

2. Select SNAME WHERE SNO='805' FROM sales;

3. Select SNAME, ADDRESS FROM sales WHERE S NO='502';

### **CASE BASED QUESTIONS (5 mark)**

1.MR MITTAL is using a table with following columns:

Name,Class,Stream\_id,Stream\_name

He needs to display names of students who have not been assigned any stream or have been assigned stream name that ends with "computers".

He wrote the following command, which did not give the desired result.

SELECT Name,Class From StudentsWhere Stream\_name=NULL OR

Stream\_name="%computers";

SELECT Name From StudentsWhere Stream\_name=NULL ;

SELECT Class From StudentsWhere Stream\_name="%computers";

SELECT Name From Students Where Stream\_id=NULL;

SELECT Class From Students Where Stream\_name="%computers";

Help Mr.Mittal to run the query by removing the error and write correct query.

2.Sarthak a student of class XII, created a table" Class" with following attribute

Sname, Sroll,Smark and Grade . Grade is one of the columns of this table. To find the details of students where Grades have not been entered, he wrote the following Mysql query which did not give the desired result:

SELECT Sname FROM Class WHERE Grade="Null";

SELECT Sroll FROM Class WHERE Grade="Null";

SELECT Smark FROM Class WHERE Grade="Null";

SELECT \* FROM Class WHERE Grade="Null";

SELECT Grade FROM Class WHERE Grade="Null";

Help Sarthak to run the query by removing the errors from the query and write the correct query.

### **SUMMARY**

The basic element of MYSQL SQL is: literals, datatypes, nulls and comments. Some Key feature of MYSQL are portability, secure system, scalable and support localization. A relational data model organizes the data into tables known as RELATIONS.

Table/Relation divided into two-part TUPLE/ROW and ATTRIBUTE/COLUMN. Some KEYS are used in TABLES like:

PRIMARY KEY, CANDIDATE KEY, ALTERNATE KEY and FOREIGN KEY.

SQL (STRUCTURED QUERY LANGUAGE). A non -procedural UGL used for querying upon relation databases.

SQL statement can be classified into:

Data Definition Language(DDL),

Data Manipulation Language (DML),

Transaction Control Language (TCL).

### **ANSWER KEY**

1. Answer: b

2. Answer: c

3. Answer: d

4. Answer: d

5. Answer: d

6. Answer: c

7. Answer: c

8. Answer: a

9. Answer: c

10. Answer: d

11. `char(n)`: stores a fixed length string between 1 and 255

Characters if the value is of smaller length, adds blank spaces  
some space is wasted

`varchar(n)`: stores a variable length string no blanks are added even if value is of smaller length no wastage of space

12. Degree: 5

Cardinality: 6

`MOVIEID` should be made the primary key as it uniquely identifies each record of the table

13. `CREATE DATABASE MYEARTH;`

`CREATE TABLE CITY`

(

`CITYCODE CHAR (5) PRIMARY KEY,`  
`CITYNAME CHAR (30),`  
`SIZE INT,`  
`AVGTEMP INT,`  
`POPULATIONRATE INT,`  
`POPULATION INT`

);

14. A `HAVING` clause is like a `WHERE` clause, but applies only to groups as a whole (that is, to the rows in the result set representing groups), whereas the `WHERE` clause applies to individual rows.

15. Primary key is a set of one or more attributes that can uniquely identify tuples within the relation. For example, in the following table `ITEM`, the column `Ino` can uniquely identify each row in the table, hence `Ino` is the primary key of the following table.

Ino	Name	Quantity
1	Pen	220
2	Cd	530
3	Dvd	450

16. Primary key is a minimal super key, so there is one and only one primary key in any relationship but there is more than one candidate key can take place. Candidate key's attributes can contain a `NULL` value which opposes to the primary key. For example, `Student{ID, First_name, Last_name, Age}`

17. Cardinality refers to the number of tuples/rows in a table whereas, Degree refers to the number of attributes/columns in a table.

18. `DDL(data definition language)`: Deals with the structure(create,remove,or modify) of databases and tables. E.g. `Create,Drop ,alter`.

`DML(data manipulation language)`:use to manipulate data/values within table e.g. `insert,delete,update`

19. An attribute value is an attribute name paired with an element of that attribute's domain, and a tuple is a set of attribute values in which no two distinct elements have the same name.

20. `DROP` is a Data Definition Language, `DDL` command and is used to remove named elements of schema like relations/table, constraints or entire schema.

Removes an existing view from a database. `DROP VIEW` statement is used to remove a view or an object view from the database

21. `ALTER TABLE Item`

`ADD (Discount INT);`

`DDL: DROP TABLE, ALTER TABLE`

`DML: INSERT INTO, UPDATE...SE`

22.

(1) 8, Mumbai

Delhi

Moscow

(2) MAX DOB (07/05/1993) ,MIN DOB(08/12/1995)

(3) sanal F

store M

23. 1. 29

2. 19-

3.

Jul-2021

T006Console Table 17-Nov-2019 15000 12

24. (a) SELECT AVG(SALARY)

FROM EMPLOYEE

GROUP BY DEPTID;

(b) SELECT NAME, DEPTNAME

FROM EMPLOYEE, DEPARTMENT

WHERE EMPLOYEE.DEPTID= DEPARTMENT.DEPTID

AND SALARY>50000;

(c ) SELECT NAME FROM EMPLOYEE

WHERE SALARY IS NULL

ORDER BY NAME;

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

Table : Stock

Id	Item	Qty
101	pen	3
102	pencil	4
102	cd	5

Id = Primary key Id and Qty = Candidates Keys

26. A table can have multiple choices for a primary key but only one can be set as the primary key. All the keys which are not primary key are called an Alternate Key.

Example: In this table, StudID, Roll No, Email are qualified to become a primary key.

Table : Student

StuID	Roll No	Sname	Email
101	210	Ajay Kumar	Ajay04@
102	220	Vijay Kumar	Vijay09@
102	240	Anil	Anil11@

StuID = Primary key Id and Roll No & Email = Alternate Keys

27. A primary key is a column -- or a group of columns -- in a table that uniquely identifies the rows of data in that table. For example, in the table below, CustomerNo, which displays the ID number assigned to different customers, is the primary key.

Table : Customer

Id No	Customer_Name	Dob
B001	Smith	28/03/1976
C002	Corth	23/09/1967
D007	John	03/02/1985

Id No = Primary key

28. Alternate Key: In a table when more than one column is used for the identification of tuples in the table, then that is an alternate key.

Each table has only one primary key, although there are several choices.  
All the columns in the table other than the primary key act as an alternate key for that table.

Candidate Key: This key is also used to identify the rows in a table.  
It contains a group of attributes that can be used to identify a particular row.  
Every table must have at least one column as a candidate key.  
In the candidate key column it does not contain any NULL value.

29. i

ANAME	ADDRESS
Nirja Singh	Bangalore
Rohan Gupta	Chennai
Ali Reza	Hyderabad
Rishabh Jain	Chennai
Simran Kaur	Chandigarh

- ii. Nirja Singh  
iii. Bangalore Chennai Hyderabad Chandigarh

30. i ) table Altered.

s.no primary key.

ii) Gurdeep Kaur

iii) 

Ram Gupta	Chennai
-----------	---------

## CASE BASE

1.

SELECT Name,Class From Students Where Stream\_name IS NULL OR Stream\_nameLIKE"%computers";

SELECT Name From StudentsWhere Stream\_name IS NULL ;

SELECT Class From Students Where Stream\_nameLIKE"%computers";

SELECT Name From Students Where Stream\_id IS Null;

SELECT Class From Students Where Stream\_nameLIKE"%computers";

2.

SELECT Sname FROM Class WHERE Grade IS "Null";

SELECT Sroll FROM Class WHERE Grade IS "Null";

SELECT Smark FROM Class WHERE Grade IS "Null";

SELECT \* FROM Class WHERE Grade IS "Null";

SELECT Grade FROM Class WHERE Grade IS "Null";

**Topics:** Select, Operators(mathematical, relational and logical), aliasing, distinct clause, where clause, in, between, order by, group by, having clause, meaning of null, is null, is not null, like, update command, delete command, aggregate functions(max, min, avg, sum , count), joins: Cartesian product on two tables, equi join and naturaljoin.

Insert, Update and Delete

## Queries:

To retrieve information from a database we can query the databases. SQL SELECT statement is used to select rows and columns from a database/relation

### SELECT Command

This command can perform **selection** as well as **projection**.

**Selection:** This capability of SQL can return you the tuples form a relation with all the attributes.

**Projection:** This is the capability of SQL to return only specific attributes in the relation.

- Select data from Table using Selectstatement

mysql> SELECT <Col Names> FROM <Table Name> WHERE <Condition>;

```
mysql> Select * from Student Where city='Barabanki';
+-----+-----+-----+-----+
| ROLL | NAME | AGE | CLASS | CITY
+-----+-----+-----+-----+
| 1001 | Pankaj | 6 | I | Barabanki |
| 1002 | Sunita | 7 | II | Barabanki |
| 1003 | Mukesh | 5 | I | Barabanki |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

- Selecting/displaying entire data from the table

SELECT \* fromtablename;

```
mysql> Select * from Student;
+-----+-----+-----+-----+
| ROLL | NAME | AGE | CLASS | CITY
+-----+-----+-----+-----+
| 1001 | Pankaj | 6 | I | Barabanki |
| 1002 | Sunita | 7 | II | Barabanki |
| 1003 | Mukesh | 5 | I | Barabanki |
| 1004 | Ramesh | 8 | II | Lucknow |
| 1005 | Yogesh | 9 | III | Kanpur |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

- Removing duplicate value from the column using distinct clause  
Table :Student

Roll_no	Name	Class	Marks	City
101	Rohan	XI	400	Chennai
102	Aneeta	XII	390	Bengaluru
103	Pawan Kumar	IX	298	Mysore
104	Rohan	IX	376	Mangalore
105	Sanjay	VII	240	Mumbai
113	Anju	VIII	432	Delhi

**SELECT DISTINCT column\_name FROM tablename;**

**DISTINCT keyword is used to restrict the duplicate rows from the results of a SELECT statement.**

e.g. **SELECT DISTINCT name FROMstudent;**

**The above command returns,**

<b>Name</b>
Roha
n
Anee
ta
Pawan Kumar

**Display all data from column using all clause**

- **SELECT ALL column\_name FROMtablename;**
- **SELECT \* FROM student;** command will display all the tuples in therelation student
- **SELECT \* FROM student WHERE Roll\_no<=102;**

**The above command display only those records whose Roll\_no less than or equal to 102.**

**Select command can also display specific attributes from a relation.**

- **SELECT name, class FROMstudent;**

**The above command displays only name and class attributes from student table.**

- **SELECT count(\*) AS "Total Number of Records" FROMstudent;**

**Displays the total number of records with title as "Total Number of Records" i.e an alias We can also use arithmetic operators in select statement, like**

- **SELECT Roll\_no, name, marks+20 FROMstudent;**
- **SELECT name, (marks/500)\*100 FROM student WHERE Roll\_no>103;**

### **Conditions based on a range**

**SQL provides a BETWEEN operator that defines a range of values that the column value must fall for the condition to become true.**

e.g. **SELECT Roll\_no, name FROM student WHERE Roll\_no BETWENN 100 AND 103;**

**The above command displays Roll\_no and name of those students whose Roll\_no lies in the range 100 to 103 (both 100 and 103 are included in the range).**

### **Conditions based on a list**

**To specify a list of values, IN operator is used. This operator select values that match any value in the given list.**

e.g. **SELECT \* FROM student WHERE city  
IN ('Bengaluru','Delhi','Chennai');**

**The above command displays all those records whose city is either Bengaluru or Delhi or Chennai**

### **ORDER BY Clause**

**ORDER BY clause is used to display the result of a query in a specific order(sorted order).**

**The sorting can be done in ascending or in descending order. It should be kept in mind that the actual data in the database is not sorted but only the results of the query are displayed in sorted order.**

e.g. **SELECT name, city FROM student ORDER BYname;**

The above query returns name and city columns of table student sorted by name in increasing/ascending order.

e.g. **SELECT \* FROM student ORDER BY cityDESC;**

It displays all the records of table student ordered by city in descendingorder. **Note:- If order is not specifies that by default the sorting will be performed in ascendingorder.**

### **GROUP BY Clause**

The GROUP BY clause can be used in a SELECT statement to collect data across multiple records and group the results by one or more columns.

The syntax for the GROUP BY clause is:

**SELECT column1, column2, ... column\_n, aggregate\_function  
(expression) FROM tables  
WHERE conditions**

**GROUP BY column1, column2, ... column\_n;**

*aggregate\_function*can be a function such as SUM, COUNT, MAX, MIN, AVG etc.

e.g   **SELECT name, COUNT(\*) as "Numberof  
employees"   FROM student WHERE  
marks>350 GROUP BYcity;**

### **HAVING Clause**

The HAVING clause is used in combination with the GROUP BY clause. It can be used in a SELECT statement to filter the records that a GROUP BY returns.

The syntax for the HAVING clause is:

**SELECT column1, column2, ... column\_n, aggregate\_function  
(expression) FROM tables**

### **WHERE predicates**

**GROUP BY column1, column2, ...  
column\_n HAVING condition1 ...  
condition\_n;**

e.g   **SELECT SUM(marks) as "Total marks"  
FROMstudent**

**GROUP BY department HAVING  
SUM(sales) > 1000;**

Note: select statement can contain only those attribute which are already present in the group by clause.

## **MODIFYING DATA IN TABLES**

❖ Syntax: UPDATE  
,TABLENAME> SET  
<COLNAME> = <VALUE>;

Write a command to modify the salary of all employee by increasing it with 5000.

**UPDATE EMP  
SET SALARY = SALARY + 5000;**

## **❖ DELETING DATA FROMTABLES**

**Syntax: DELETE FROM <TABLE NAME> WHERE <SEARCH CONDITION>;**

**Write a command to delete all rows from EMP whose deptno is 10. DELETE FROM EMP WHERE DEPTNO = 10;**

## **❖ ALTERING TABLES**

**Syntax: ALTER TABLE <TABLENAME>; ADD/MODIFY/CHANGE <COLNAME><DATATYPE>;**

**Write a command to add a new column PHNO in table emp.**

```
ALTER TABLE EMP
ADD PHNO INT;
```

**Write a command to modify column Job the table EMP , change the width of it to 30.**

```
ALTER TABLE EMP
MODIFY JOB VARCHAR(30);
```

**Write a command to change the existing column name ENAME to EMPNAME in table EMP.**

```
ALTER TABLE EMP
CHANGE ENAME EMPNAME VARCHAR(30);
```

## **❖ DROPPING TABLES**

**Syntax :**

```
DROP TABLE [IF EXISTS] <TABLE NAME>;
```

**Write a command to drop table emp. DROP TABLE IF EXISTS EMP;**

### **Performing simple calculation in the query Syntax:**

**Select 2+4;**

**All operation can be perform such as addition (+), subtraction (-), multiplication (\*), division (/) and remainder (%)**

**Number of row in result depend on total number of rows in the table.**

**> Performing calculation on column**

**Syntax:SELECT column1 operator value, column2.... FROM Tablename**

### **> Working with nullvalue**

**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

**Here, comm column contains null value which is replaced by „N/A”.**

## **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.**

## **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 –  
N  
o  
t  
i  
n

The IN operator allows you to specify multiple values in a WHERE clause. The IN operator is a shorthand for multiple OR conditions.

## **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
  - min() – return minimum value from the set of values
  - avg() –return average value in set of non-null values
  - sum() - Return the summation of all non-NULL values of the set of values.

### **count() - Return the number of rows in a group**

- count(\*) – return number of rows, including rows with NULL clause

```
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
- count (distinct column\_name) – return number of rows with duplicates removed

In above example there are 14 row in the EMP table, but distinct clause only consider unique value.

## JOIN

A JOIN is a query through which we can extract queries from two or more tables. It means, it combines rows from two or more tables. Rows in one table can be joined to rows in another table according to common values existing in corresponding columns.

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 EMPLOYEES, DEPARTMENTS and JOBGRADES that stored related information, all the examples on join will be explained with help of these following three tables.

### 1. EQUI-JOIN

In an EQUI-JOIN operation, the values in the columns are being joined and compared for equality. All the columns in the tables being joined are included in the results, e.g. Two tables EMPLOYEES and DEPARTMENTS are given below:

EMPLOYEES	
Employee_id	Department_id
200	10
201	20
202	20
124	50
141	50
103	60
102	40
105	30
106	30

DEPARTMENTS	
Department_id	Department_name
10	Administration
20	Marketing
30	I.T.
40	Accounting
50	Shipping
60	Development

To determine an employee's department name you compare the value Department\_id column in the EMPLOYEES table with Department\_id values in the DEPARTMENTS table.

The relationship between the EMPLOYEES and

DEPARTMENTS tables is an equi-join that is, values in the Department\_id column on both tables must be equal.

To determine employee's department name, we need to write following query:

```
SELECT EMPLOYEES.Employee_id, DEPARTMENTS.Department_name FROM
EMPLOYEES, DEPARTMENTS
```

```
WHERE EMPLOYEES.Department_id = DEPARTMENTS.Department_id;
```

**Output**

EMPLOYEES.Employee_id	DEPARTMENTS.Department_name
200	Administration
201	Marketing
202	Marketing
124	Shipping
141	Shipping
103	Development
102	Accounting
105	I.T.
106	I.T.

## Non-EquiJoin

A non-equi join is a join condition containing something other than an equality operator, e.g. there are two given tables **EMPLOYEES** and **JOB\_GRADES**

EMPLOYEES		JOB_GRADES		
Last_name	SALARY	GRA	LOWEST_SAL	HIGHEST_SAL
Malik	24,000	A	1000	2999
King	17,000	B	3000	5999
Honold	9,000	C	6000	9999
Davies	3,100	D	10000	14999
Mougos	5,800	E	15000	24999
Ernst	6,000	F	25000	40000

The relationship between the **EMPLOYEES** table and **JOB\_GRADES** table has an example of a non-equi join. A relationship between the two tables is that the **SALARY** column in the **EMPLOYEES** table must be between the values in **LOWEST\_SAL** and **HIGHEST\_SAL** columns of the **JOB\_GRADES** table. The relationship is obtained using an operator other than equals (=).

To determine the employees grade according to salary, we need to write following query:

```
SELECT e.Last_name, j.GRA FROM EMPLOYEES e, JOELGRADES j WHERE
e.SALARY BETWEEN j.LOWEST_SAL AND j.HIGHEST_SAL;
```

### Output

e. Last_name	j. GRA
Malik	E
King	E
Honold	C
Davies	B
Mougos	B
Ernst	C

## 2. NaturalJoin

Usually the result of an equi-join contains two identical columns. Here by restarting the query, we can eliminate one of the two identical columns. It is known as Natural Join.

We can also join two tables using the natural join using NATURAL JOIN clause. `SELECT * FROM <table1> NATURAL JOIN <table2>;` e.g. there are two given tables **FOOD** and **COMPANY**:

FOOD			COMPANY		
Item_id	Item_Name	C_id	C_id	C_Name	C_City
I1	Rice	14	12	Aakash Food	Delhi
I2	Cake	12	14	Order Today	Meerut
I3	Juice	15	15	Foodies	Agra
			16	Food Now	Kota

The relationship between the **FOOD** table and **COMPANY** table has an example of a Natural Join. To get all the unique columns from **FOOD** and **COMPANY** tables, the following sql statement can be used.

e.g. SELECT \* FROM FOODNATURAL JOIN COMPANY;

### Output

C_id	Item_id	Item_Name	C_Name	C_City
14	I1	Rice	Order Today	Meerut
12	I2	Cake	Aakash Food	Delhi
15	I3	Juice	Foodies	Agra

### Cartesian Product

The cartesian product is a binary operation and is denoted by  $(x)$ . The degree of new relation is the sum of the degrees of two relations on which Cartesian product is operated. The number of tuples, of the new relation is equal to the product of the number of tuples, of the two relations on which Cartesian product is performed.

e.g. if  $A = \{1, 2, 3\}$  and  $B = \{a, b, c\}$ , find  $A \times B$ .

$$A \times B = \{(1, a), (1, b), (1, c), (2, a), (2, b), (2, c), (3, a), (3, b), (3, c)\}$$

In SQL the CROSS JOIN or CARTESIAN JOIN is used to produce the cartesian product of two tables.

The Cartesian product is a basic type of join that matches each row from one table to every row from another table.

e.g. Consider the following EMPLOYEES and DEPARTMENTS tables:

EMPLOYEES			DEPARTMENTS	
EMP_name	EMP_id		Dpt_id	
Naveen	101		10	
Vikash	102		20	
			30	

To get the cartesian product, the following sql statement can be used:

SELECT EMP\_name, EMP\_id FROM EMPLOYEES CROSS JOIN DEPARTMENTS;

### Output

EMP_name	EMP_id
Naveen	101
Naveen	101
Naveen	101
Vikash	102
Vikash	102
Vikash	102

### 1 MARK QUESTIONS

#### **1. What is connection? What is its role?**

Ans. A Connection (represented through a connection object) is the session between the application program and the database. To do anything with database, one must have a connection object

#### **2. Which package must be imported in Python to create a database connectivity application?**

Ans. There are multiple packages available through which database connectivity applications can be created in Python. One such package is mysql.connector.

#### **3. Which function is used to get multiple records retrieved as the result of SQL query executed?**

Ans. fetchmany()

**4. Which method is used to close the open database connection?**

Ans. <connectionobject>.close()

**5. To make the changes made by any SQL Queries permanently in database, which function is used after execution of the query ?**

Ans. <connectionobject>.commit()

**6. Which function is used to execute an SQL query from within a Python program?**

Ans. execute()

**7. Identify the name of connector to establish bridge between Python and MySQL**

- a) mysql.connection
- b) connector
- c) mysql.connect
- d) mysql.connector

Ans. d. mysql.connector

**8. In the following connection string: Identify the elements:**

connect(<<1>> \_\_\_\_\_ =127.0.0.1, \_\_\_\_\_ <<2>> \_\_\_\_\_ ="root", \_\_\_\_\_ <<3>> \_\_\_\_\_ ="admin")

- a. <<1>> = User, <<2>> = password, <<3>> =host
- b. <<1>> = host, <<2>> = user, <<3>> =password
- c. <<1>> = host, <<2>> = password, <<3>> =user
- d. <<1>> = IP, <<2>> = user, <<3>> = password**

Ans. b. <<1>> = host, <<2>> = user, <<3>> = password

**connect(host= 127.0.0.1, user= 'root', password= 'admin')**

**9. 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 one at a time?**

- a) ResultSet
- b) Cursor
- c) Container
- d) Table

Ans. b. Cursor

**10.Which attribute of cursor is used to get number of records stored in cursor (Assuming cursor name is mycursor)?**

- a) mycursor.count
- b) mycursor.row\_count
- c) mycursor.records
- d) mycursor.rowcount

Ans. d. mycursor.rowcount

### **2 MARKS QUESTIONS**

**1. Write the code to create the connection in which database's name is Python, name of host, user and password can taken by user. Also, print that connection?**

Ans.import mysql.connector

```
mycon = mysql.connector.connect(host = "localhost", user = "root", passwd = "tiger",
database = "Python")
print(mycon)
```

**3. Explain the following results retrieval methods with examples-**

**fetchone () rowcount ()**

Ans. fetchone() :- The fetchone() method will return only one row from the result set in the form of tuple containing a record.

rowcount() :- cursor.rowcount() that always return how many records have been retrieved so far using any of the fetch..() methods.

**3. Write a small python program to insert a record in the table books with attributes (title ,isbn).**

```
Ans. import mysql.connector as Sqlator
conn =sqlator.connect(host="localhost",user="root",passwd="",database="test")
cursor=con.cursor()
query="INSERT into books(title,isbn) values('{}'{}'.format('Neelesh','5143')
cursor.execute(query)
con.close()
```

**4. Differentiate between fetchone() and fetchall() methods.**

Ans. fetch(): It returns the next row from the result set as tuple. If there are no more rows to retrieve, None is returned.

fetchall():It fetches all the rows of a query result. It returns all the rows as a list of tuples.

An empty list is returned if there is no record to fetch.

**5. Consider the following Python code 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
- c) con.commit() function is missing
- d) commit() function is missing

Ans. c. con.commit() function is missing

### **3 MARKS QUESTIONS**

**1. Avni is trying to connect Python with MySQL for her project. Help her to write the python statement on the following:-**

**(i) Name the library, which should be imported to connect MySQL with Python.**

**(ii) Name the function, used to run SQL query in Python.**

**(iii) Write Python statement of connect function having the arguments values**

**as : Host name :192.168.11.111**

User : root

Password: Admin

Database : MYPROJECT

Ans. (i) import mysql.connector

(ii) execute (<sql query >)

```
(iii)mysql.connector.connect(host="192.168.11.111",user="root",passwd="Admin",databa
```

```
se="MYPROJECT")
```

**2. The given program is used to connect with MySQL abd show the name of the all the record from the table "stmaster" from the database "oraclenk". You are required to complete the statements so that the code can be executed properly.**

```
import _____.connector_pymysql
dbcon=pymysql._____ (host="localhost", user="root",
_____= "sia@1928")
if dbcon.isconnected()==False
 print("Error in establishing connection:")
cur=dbcon._____()
query="select * from stmaster"
cur.execute(_____
resultset=cur.fetchmany(3)
for row in resultset:
 print(row)
dbcon._____()
```

Ans. import mysql.connectoraspymysql

```
dbcon=pymysql.connect(host="localhost", user="root", passwd="sia@1928")
if dbcon.isconnected()==False
 print("Error in establishing connection:")
cur=dbcon.cursor()
query="select * from stmaster"
cur.execute(query)
resultset=cur.fetchmany(3)
for row in resultset:
 print(row)
dbcon.close()
```

**3. Explain various database operations one can perform using MySQL-Python connectivity.**

Ans. INSERT, READ, UPDATE, DELETE, ROLLBACK (explain in detail )

4. Consider the information stored in the table: EMP

EMPNO	ENAME	DEPT	SALARY
1	ALEX	MUSIC	60000
2	PETER	ART	67000
3	JOHNY	WE	55000
4	RAMBO	P&HE	48000

Following python code is written to access the records of table : EMP, What will be the output of following code?

```
Assume All basic setup related to connection and cursor creation is already done
query="select * from emp"
mycursor.execute(query)
results = mycursor.fetchone()
results = mycursor.fetchone()
results = mycursor.fetchone()
d = int(results[3])
print(d*3)
```

Ans. 165000

**5. Find the output:**

```
import mysql.connector as mys
mycon = mys.connect(host='localhost',user='root',passwd='admin',database='company')
mycursor = mycon.cursor()
mycursor.execute("select * from emp")
mydata = mycursor.fetchone()
nrec = mycursor.rowcount
print("Total records fetched so far are",nrec)
mydata = mycursor.fetchone()
nrec = mycursor.rowcount
print("Total records fetched so far are",nrec)
mydata = mycursor.fetchmany(2)
nrec = mycursor.rowcount
print("Total records fetched so far are",nrec)
```

Ans. Total records fetched so far are 1

Total records fetched so far are2

Total records fetched so far are4