Name of the chapter : **Introduction to Python**

Topic Covered

- Basics of Python programming,Python interpreter-interactive and script mode, the structure of a program,
- Indentation,identifiers,keywords,constants,variables,types of operators,precedence of operators,data
- Types,mutable and immutable data types,statements,expressions,evaluation and comments, input and output statements
- Data type conversion,debugging.
- Control Statements:if-else,**if-elif-else, while loop,**forloop
- Lists: list operations-creating,initializing,traversing and manipulating lists,list methods and built-infunctions.– len(),list(),append(),insert(), count(),index(),remove(), pop(), reverse(), sort(), min(),max(),sum()
- Dictionary: concept to fkey-value pair, creating,initializing,traversing, updating and deleting elements,dictionary methods and built-infunctions.– dict(), len(), keys(), values(), items(), update(), del(), clear()

Key Points

Python interpreter To execute a program in a high-level language by translating it one line at a time interactive mode, A way of using the Python interpreter by typing commands and expressions at the prompt.
script mode, A way of using the Python interpreter to read and execute statements in a script.
In Python, everything is an object. For example, numbers, strings, functions, classes, and modules are all objects. Every Python object has three core characteristics that define it at a foundational level. These characteristics are: Value ,identity

In Python, variables don't have an associated type or size, as they're labels attached to objects in memory

Python **objects** are concrete pieces of information that live in specific memory positions on your computer.

An object's value is probably the only characteristic that you'd want to change in your code. An object that allows you to change its values without changing its identity is a **mutable**

In contrast, an object that doesn't allow changes in its value is an **immutable** object.

Control Statements- It allows programmers to control the execution flow of a program or one of its sections.
if statement- if statement consists of a Boolean expression followed by one or more statements. If the condition is True, the statements under if statements are executed.
if else-An if else statement consists of a Boolean expression followed by one or more statements. If the condition is True, the statements under if statements are executed. If the condition is false, statements under else part is executed
if elif- An if statement followed by one or more elif Statements, that consists of Boolean expressions and then followed by an optional else statement, which executes when all the condition becomes false.

Nested if - An if statement inside another if or elif statement(s).
Loop- executes a statement or group of statements multiple times
while loop- It consists of a Boolean expression written along with while keyword followed by one or more statements which will be executed as long as condition is True
for loop- Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable
break - It is used to terminate the loop.
c**ontinue** -It is used to skip all the remaining statements in the loop and move controls back to the top of the loop.
pass-This statement does nothing. It can be used when a statement is required syntactically but the program requires no action.

Definition :

A list is a data structure in Python that is a mutable, or changeable, ordered sequence of elements.
Example:- L1=[10,25,100,500], L2=[1, 2, 2.5, 10.0, 'a', 'b'], L3=[1,[2,3],4]

List Creation:

L1=[] or L2=list( ) # To create empty list
L3=[10,25,100,500] # To create and initialize list
L4=eval(input("Enter elements of the list")) #to create and initialize by user input

List Traversing and Manipulation:

Every element of the list has an unique sequential index(position) starting from 0
List elements can be accessed and manipulated by index.
if LST=[3,6,9,12,15] then
LST[1] ---->refers second element, i.e 6
LST[3]=20 ----> modifies the 4th element as 20.

To traverse and display all elements of the list:-

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

List Operators:

Concatenation (+)

Joins two lists. For example if L1=[1,2,3], L2=[4,5] then L1+L2 ---> [1,2,3,4,5]

Replication (*)

Replicates the list given number of time.
For example , L1*3 ----->[1,2,3,1,2,3,1,2,3]

Membership (IN / NOT IN)

Checks if an element is present or not.
For example : 2 in L1---> True 2 in L2--->False

Comparison (==, !=, >, <, >=, <=)

Compares two lists element by element
For example:
L1==L2---->False L1!=L2= True L1>L2-----> False

Slice (:)

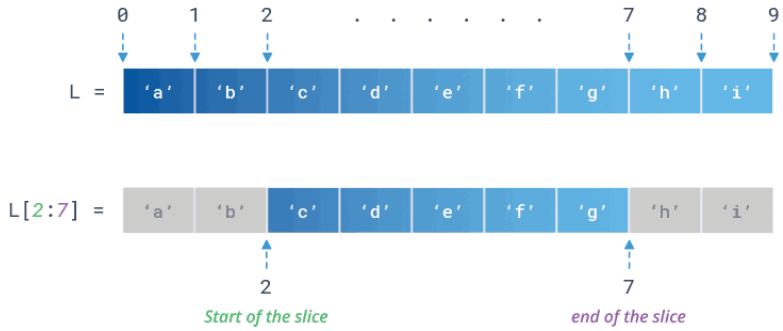
To access a range of items in a list, you need to slice a list. One way to do this is to use the simple slicing operator ":".
Syntax:

```
L[start:stop:step]
```

Start position End position The increment

Note: If step is omitted, default step in 1.
Example:

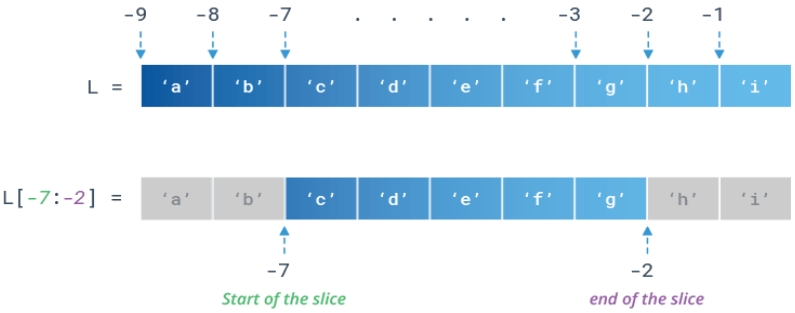
```
L = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i']
print(L[2:7])
# Prints ['c', 'd', 'e', 'f', 'g']
```



Note: Last index is always excluded.

Negative Index:

```
L = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i']
print(L[-7:-2])
# Prints ['c', 'd', 'e', 'f', 'g']
```



Built in Functions

len()

returns the no of size/no of elements. len(L1)---->3 , len(L2)----->2

min()

returns minimum element min(L1) ----->1

max()

returns maximum element max(L2) ----->5

sum()

returns sum of elements sum(L2) ----->9 sum(L1)----->6

List Functions

append()

adds an element at the end of the list
L1.append(10) ------> [1,2,3,10]

insert()

insert an element at a given index
L2.insert(1,2.5) ------->[4,2.5,5]

count()
returns the frequency of an element
Val=[2,4,6,4,7,3,4]
Val.count(4)------>3
index()
returns the index of an element
Val.index(6)------>2
remove()
delete the element with a given value
Val.remove(7) -----> [2,4,6,4,3,4]
pop()
delete the element with a given index . If no index is given, last element is deleted
Val.pop(2) -----> Val=[2,4,4,7,3,4]
reverse()
Val.reverse( )----->[4,3,7,4,4,2]
arranges the elements of the list in a reverse order
sort()
arranges the elements of a list in ascending order.
cars = ['Ford', 'BMW', 'Volvo']
cars.sort() ----> ['BMW', 'Ford', 'Volvo']
clear( )
deletes all elements of the list
Num=[1,4,7,9]
Num.clear() -----> Will give empty list
extend()
Merge the elements of a list in the current list
L1.extend(L2) ----->[1,2,3,4,5]
L2.extend(L1) ------>[4,5,1,2,3]
Dictionary

- Dictionary has some similarities with string, list and tuple but it is different in terms of storing and accessing an element. String, list and tuple are sequences whereas a dictionary is a mapping. Rather than having an index associated with each element, Dictionaries in Python have a **key** associated with each element. Python Dictionaries are a collection of **key value pairs**.
- In other ways, you can think keys as user defined indices.
- In English Dictionary, we search any word for meaning associated with the word. Here Word is the **Key** and meaning is the **value**.

Dictionary Creation:

- Syntax:
  <dictionay_name>={<key1>:<value1>, <key2>:<value2>, ……..
  <keyn>:<value n>}
- Example:
  Price={ 'Redmi Note8': 10500, 'Galaxy A70s': 25900,
  'OppoA31':12490}
  Here 'Redmi Note8', 'Galaxy A70s' and 'Oppo A31' are Keys and 10500, 25900, 12490 are values.

- Dict1={} #Empty Dictionary
- TeacherCount={'PGT':10, 'TGT':7, 'PRT':5}

- Note: Keys of a Dictionary must be of immutable types, such as:
  * A Python string
  * A number.
  * A tuple (containing only immutable types)

  If we try to give a mutable type as key, Python will give an error.

Accessing elements of a Dictionary:

- We need a key to access any element of a Dictionary likewise in lists, strings and tuples we used index to access any element.
- Example:

```
>>> TeacherCount={'PGT':10, 'TGT':7, 'PRT':5}
>>> TeacherCount{'PGT': 10, 'TGT': 7, 'PRT': 5}
>>> TeacherCount['PGT']
10
```

- Note: Attempting to access a key that doesn't exist causes an error

Traversing a Dictionary:

- Traversal of a collection means accessing and processing each element of it.
- for loop is efficient to traverse any collection and sequence. for loop will get every key of Dictionary and we can access every element of the Dictionary based on the keys.
- **Example:**

```
>>> TeacherCount={'PGT':10, 'TGT':7, 'PRT':5}
>>> TeacherCount
{'PGT': 10, 'TGT': 7, 'PRT': 5}
>>> for i in TeacherCount:
                print('Key is ',i, 'Value is ',TeacherCount[i])
Key is  PGT Value is  10
Key is  TGT Value is  7
Key is  PRT Value is  5
```

Accessing Key and Values Simultaneously:

- Accessing all keys in a Dictionary in one go:

  <dictionary>.keys()

  Example:

```
 >>> TeacherCount={'PGT':10, 'TGT':7, 'PRT':5}
>>> TeacherCount
{'PGT': 10, 'TGT': 7, 'PRT': 5}
>>> TeacherCount.keys()
dict_keys(['PGT', 'TGT', 'PRT'])
```

- Accessing all values in a Dictionary in one go:

  <dictionary>.values()

  Example:

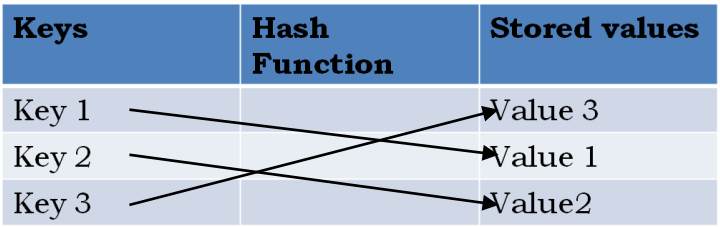
```
 >>> TeacherCount.values()
dict_values([10, 7, 5])
```

Characteristics of a Dictionary:

- **Unordered Set:**

  A dictionary is a unordered set of key:value pair.

- **Not a Sequence:**
  Unlike a string, list and tuple, a dictionary is not a sequence because it is unordered set of elements. The sequences are indexed by a range of ordinal numbers. Hence, they are ordered, but a dictionary is an unordered collection.
- **Indexed by keys, Not numbers:**
  Dictionaries are indexed by keys. Keys are immutable type. But the values of a dictionary can be of any type.
- **Keys must be unique:**
  Each keys of a Dictionary must be unique. However two unique keys can have same values.
- **Mutable:**
  Like lists, dictionaries are mutable. We can change the value of a key in **place**.
- **Internally stored as Mappings:**
  Internally, the key:value pairs of a dictionary are associated with one another with some internal function (called hash-function). This way of linking is called mapping.



Multiple ways of Creating Dictionaries:

1. Initializing a Dictionary:

In this method all the **key:value** pairs of a dictionary are written collectively, separated by commas and enclosed in curly braces.

Example:

```
T20Cricketer={'Name':'Virat', 'Runs':2794, 'Age':31}
```

2. Adding key:value pairs to an empty Dictionary:

In this method, first we need to create an empty dictionary and then keys and values are added to it one pair at a time.

Two ways of creating empty dictionary:

- T20Cricketer={}

and

- T20Cricketer=dict()

Next step is to add key:value pairs, one at a time as per syntax given below:

<dictionary>[<key>]=<value>

Example:

```
>>> T20Cricketer['Name']='Virat'
>>> T20Cricketer['Runs']=2794
>>> T20Cricketer['Age']=31
>>> T20Cricketer
{'Name': 'Virat', 'Runs': 2794, 'Age': 31}
```

3. Creating a Dictionary using dict():

(i) Specify Key:Value pairs as keyword arguments to dict() function:

Example:

```
>>> T20Cricketer=dict(Name='Virat', Runs=2794, Age=31)
>>> T20Cricketer
{'Name': 'Virat', 'Runs': 2794, 'Age': 31}
```

(ii) Specify comma-separated Key:Value pairs:

Example:

```
>>> T20Cricketer=dict({'Name':'Virat', 'Runs':2794, 'Age':31})
>>> T20Cricketer
{'Name': 'Virat', 'Runs': 2794, 'Age': 31}
```

(iii) Specify Key:Value pairs separately in form of sequences:

In this method, one list or tuple of individual key value pair is passed as argument to dict().

Example1:

```
>>>T20Cricketer=dict([['Name','Virat'],['Runs',2794],
                                        ['Age',31]])
>>> T20Cricketer
{'Name': 'Virat', 'Runs': 2794, 'Age': 31}
```

Example 2:

```
>>>T20Cricketer=dict((('Name','Virat'),('Runs',2794),
                    ('Age',31)))
>>> T20Cricketer
{'Name': 'Virat', 'Runs': 2794, 'Age': 31}
```

Example3:

```
>>>T20Cricketer=dict((['Name','Virat'],['Runs',2794],
                    ['Age',31]))
>>> T20Cricketer
{'Name': 'Virat', 'Runs': 2794, 'Age': 31}
```

Example 4:

```
>>>T20Cricketer=dict([('Name','Virat'),('Runs',2794),
                    ('Age',31)])
>>> T20Cricketer
{'Name': 'Virat', 'Runs': 2794, 'Age': 31}
```

- **Adding Elements to Dictionary:**

<dictionary>[<key>]=<value>

Example:

```
>>> T20Cricketer={'Name':'Virat','Runs':2794,
                  'Age':31}
>>> T20Cricketer['Country']='India'
>>> T20Cricketer
{'Name': 'Virat', 'Runs': 2794, 'Age': 31, 'Country':'India'}
```

N.B: Key to be added must not exist in dictionary and must be unique. If the Key already exists, then it will change the value of existing key and no new entry will be added to dictionary.

- **Updating Existing Elements in a Dictionary:**

<dictionary>[<key>]=<value>

Example:

```
>>> T20Cricketer={'Name':'Virat','Runs':2794,
                  'Age':31}
>>> T20Cricketer
{'Name': 'Virat', 'Runs': 2794, 'Age': 31}
>>> T20Cricketer['Name']='Rohit'
>>> T20Cricketer
{'Name': 'Rohit', 'Runs': 2794, 'Age': 31}
```

N.B: Key must exist in the dictionary otherwise new entry will be added to dictionary.

- **Deleting Elements from a Dictionary using del:**

  del <dictionary>[<key>]

  Example:

```
>>> T20Cricketer={'Name':'Virat','Runs':2794,
                  'Age':31}
>>> T20Cricketer
{'Name': 'Virat', 'Runs': 2794, 'Age': 31}
>>> del T20Cricketer['Age']
>>> T20Cricketer
{'Name': 'Virat', 'Runs': 2794}
```

N.B: Key must exist in the dictionary otherwise Python gives KeyError.

- **Checking for Existence of a Key:**

Membership operators in and not in are used to check for the existence of Keys only.

  <key> in <dictionary>

  <key> not in <dictionary>

  Example:

```
>>> T20Cricketer={'Name':'Virat','Runs':2794,
                                    'Age':31}
>>> 'Runs' in T20Cricketer
True
>>> 'Age' not in T20Cricketer
False
>>> 'Virat' in T20Cricketer
False
# Here 'Virat' is not a key
```

- **Checking for Existence of a Value:**

  <value> **in** <dictionary>.values()

  <value> **not in** <dictionary>.values()

  Example:

```
>>> T20Cricketer={'Name':'Virat','Runs':2794,
                  'Age':31}
 >>> 'Virat' in T20Cricketer.values()
True
```

- **Functions and Dictionary Methods:**

1. len(Dictionary):

Returns total number of elements present in the dictionary.

Example:

```
T20Cricketer={'Name':'Virat','Runs':2794,'Age':31}
>>> len(T20Cricketer)
3
```

2. Dictionary.keys():

Returns a list of keys of the dictionary.

Example:

```
>>> T20Cricketer.keys()
dict_keys(['Name', 'Runs', 'Age'])
```

3. Dictionary.values():
Returns a list of values of the dictionary.
Example:
>>> T20Cricketer.values()
dict_values(['Virat', 2794, 31])

4. Dictionary.items() :
Returns all of the items of the dictionary as a sequence of (key,value) tuples
Example:
>>> T20Cricketer.items()
dict_items([('Name', 'Virat'), ('Runs', 2794), ('Age', 31)])

5. Dictionary.update(other dictionary) :
Merges key:value pairs from the other dictionary to the original Dictionary. It updates the value of the keys if the keys exist in the original dictionary otherwise adds the key:value pair to the original Dictionary.
Example:
>>> T20Cricketer
{'Name': 'Virat', 'Runs': 2794, 'Age': 31}
>>> T20Cricketer2
{'Name': 'Rohit', 'Age': 33, 'Country': 'India'}
>>> T20Cricketer.update(T20Cricketer2)
>>> T20Cricketer
{'Name': 'Rohit', 'Runs': 2794, 'Age': 33, 'Country': 'India'}
>>> T20Cricketer2
{'Name': 'Rohit', 'Age': 33, 'Country': 'India'}

6. del <Dictionary>[<key>]
Deletes key:value pair or element of the Dictionary.
Example:
>>> T20Cricketer
{'Name': 'Rohit', 'Runs': 2794, 'Age': 33, 'Country': 'India'}
>>> del T20Cricketer['Country']
>>> T20Cricketer
{'Name': 'Rohit', 'Runs': 2794, 'Age': 33}
7. del Dictionary
Deletes entire Dictionary
Example:
>>> del T20Cricketer2
>>> T20Cricketer2
NameError: name 'T20Cricketer2' is not defined
8. Dictionary.clear()
Removes all items of the Dictionary leaving the Dictionary empty.
Example:
>>> T20Cricketer2
{'Name': 'Rohit', 'Runs': 2773, 'Age': 33}
>>> T20Cricketer2.clear()
>>> T20Cricketer2
{}

Nested Dictionary:
A Dictionary is called nested if there is at least one Dictionary as a value of key.
Example

```
>>> CSTeacher={'Name':{'Fname':'Rajat','Lname':'Bhatia'},'Desig':'PGT'}
>>> CSTeacher
{'Name': {'Fname': 'Rajat', 'Lname': 'Bhatia'}, 'Desig': 'PGT'}
>>> CSTeacher['Name']
{'Fname': 'Rajat', 'Lname': 'Bhatia'}
>>> CSTeacher['Name']['Fname']
'Rajat'
```

55 Objective Question (1 Mark)	
Q1.	What are the two modes of Python interpreter?
Ans	Interactive mode and script mode
Q2.	Identify the invalid identifier: (a) sum1 (b) _sum (c) sum@ (d) SUM
Ans	(c)
Q3.	In _____ mode of Python, we can save the program.
Ans	Script
Q4.	Which is of the following is not a constant? (a) True (b) “Hello” (c) 3.14 (d) sum
Ans	(d)
Q5.	Identify the immutable data type: (a) dictionary (b) int (c) list (d) set
Ans	(b)
Q6.	Which one of the following is the correct extension of the Python file? a) .py b) .python c) .p d) None of these
Ans	(a)
Q7.	Which is the correct operator for power(xy)? a) X^y b) X**y c) X^^y d) None of the mentioned
Ans	(b)
Q8.	Python supports dynamic typing. – True / False
Ans	True
Q9.	a = 5 > 2 What will be the data type of the variable a? (a) True (b) int (c) bool (d) None of these
Ans	(c) bool

Q10.	Which of the following is not a token : (a) // (b) "a" (c) 3.14 (d) ##
Ans	(d)
Q11	What will be the output of the following code snippet: n=3 n=4 n=n+n print(n) (a) 7 (b) 6 (c) 1 (d) 8
Ans	(d)
Q12	What will be the value of the following Python expression : 4 + 3 % 5 (a) 2 (b) 4 (c) 7 (d) Error
Ans	(c)
Q13	Which function displays the memory location of an object/variable ?
Ans	id( )
Q14	_____ spaces should be left for indentation. (a) 2 (b) 3 (c) 4 (d) 1
Ans	(c) 4
Q15	Python is case-sensitive – True / False.
Ans	True
Q16.	What keyword would you use to add an alternative condition to an if statement? a) else if b) elseif c) elif d) None of the above
Ans	c) elif
Q17.	How is a code block indicated in Python? a) Brackets b) Indentation c) Key d) None of the above
Ans	b) Indentation
Q18.	The order of execution of the statements in a program is known as: a) flow of control b) central flow c)selection d) iteration
Ans	a) flow of control

Q19.	Number of elif in a program is dependent on the _____ a) number of conditions to be checked b) number of variables in a program c) number of loops in a program d) None of the above
Ans	a) number of conditions to be checked
Q20.	An 'if' condition inside another 'if' is called ___ a) Second if b) nested if c) another if d) None of the above
Ans	b) nested if
Q21.	____ is an empty statement in Python. a) Jump b) Fail c) Empty d) Pass
Ans	d) Pass
Q22.	Which of the following symbol is used to end an 'if' statement in Python? a) Comma(,) b) Colon(:) c) Semi Colon(;) d) None of the above
Ans	b) Colon(:)
Q23.	Repetition of a set of statements in a program is made possible using _____ a) Selection Constructs b) Sequential Constructs c) Looping Constructs d) None of the above
Ans	c) Looping Constructs
Q24.	The statements in a loop are executed repeatedly as long as particular condition _____. a) remains False b) remains True c) gives error d) None of the above
Ans	b) remains True
Q25.	When the condition in loops becomes false, the loop _____ a) terminates b) begin c) restart d) none of the above
Ans	a) terminates
Q26	Consider the loop given below: for i in range(7,4,-2) : break What will be the final value of i after this loop? a) 4 b) 5 c) 7 d) -2
Ans	b) 7

Q27	Consider the loop given below: for i in range(10,5,-3) : print(i) How many times will this loop run? a) 3 b) 2 c) 1 d) Infinite
Ans	b) 2
Q28	Consider the loop given below: for i in range(3) : pass What will be the final value of i after this loop? a) 0 b) 1 c) 2 d) 3
Ans	c) 2
Q29	Consider the loop given below: for i in range(2,4) : print(i) What value(s) are printed when it executes? a) 3 b) 3 and 4 c) 2 and 3 d) 2,3 and 4
Ans	c) 2 and 3
Q30	Function range(3) is equivalent to: a) range(1,3) b) range(0,3) c) range(3,0,-1) d) range(1,3,0)
Ans	b) range(0,3)
Q31.	Suppose L=[10,20,30,40,50,60] , then what is the value of L[::2]?
Ans	[10, 30, 50]
Q32.	If L1=[‘a’,’b’,’c’] then find 2*L1
Ans	['a', 'b', 'c', 'a', 'b', 'c']
Q33.	Consider a list LST=[2,3,[1,5]] . Find the output of the statement: 1 in LST
Ans	False.
Q34.	If L=list(‘123’) then find the output of the statement : print(L)
Ans	[‘1’,’2’,’3’]
Q35.	If List1=[[‘a’,’b’,’c’],[10,20,30]] then find the value of len(List1)
Ans	2
Q36.	Consider a list LST=[10,20,30,40]. Write a statement to insert element 50 at the last position.
Ans	LST.append(50)
Q37.	Consider a list LST123=[1,2,3,4]. Write a statement to insert element 2.5 at index no 3.
Ans	LST123.insert(3, 2.5)

Q38.	If LST = 'SUMMER' then find LST[::-1]
Ans	REMMUS
Q39.	Write a ststement to create an empty list.
Ans	L=[] or L=list( )
Q40.	Which of the following function is a standard library function and not a list function? a. pop( ) b. max( ) c. extend( ) d. sort( )
Ans	b. max( )
Q41	Which function is used to merge two lists into a single list?
Ans	extend( )
Q42	Which operator will be used to make a copy of a list to another list? a. = b. == c.+ d.*
Ans	a.=
Q43	Write the result of the statement : print(list(range(5))
Ans	[0,1,2,3,4]
Q44	Suppose a list L=[1,2,3,4,5]. Write a statement to remove all the elements a make an empty list, i.e, L=[]
Ans	L.clear()
Q45	Find the output: - L=[0,[9,'a'],77.9,'KVS',['Rahul','Viki','Vijay]] print(L[:3]+L[1::-1])
Ans	[0, [9, 'a'], 77.9, [9, 'a'], 0]
Q46.	Which of the following is correct way of creating a dictionary? a) Medals={'Gold'=12,'Silver'=21,'Bronze'=32} b) Medals={'Gold':12,'Silver':21,'Bronze':32} c) Medals=['Gold':12,'Silver':21,'Bronze':32] d) Medals=('Gold':12,'Silver':21,'Bronze':32)
Ans	b) Medals={'Gold':12,'Silver':21,'Bronze':32}
Q47.	Dictionary is a _____ a) Set b) Sequence c) Mapping d) None of the options
Ans	c) Mapping
Q48.	Which one of the following statement is not True? a) Dictionary is value mutable. b) Dictionary is key immutable. c) Dictionary is a mapping. d) Dictionary is an ordered set of items.
Ans	d) Dictionary is an ordered set of items.
Q49.	Find out the odd one from the following: a) Integer b) String c) Float d) Dictionary
Ans	d) Dictionary
Q50.	Which of the following statement is wrong? a) D={1:2,3:4,4:5} b) D={[1,2]:'Tarun',[3,5]:'Komal',[4,6,7]:'Sampreet'} c) D=dict({1:'Madhu',2:'Karan',3:'Mohan'}) d) D={'Tarun': [1,2],'Komal': [3,5],'Sampreet': [4,6,7]:}
Ans	D={[1,2]:'Tarun',[3,5]:'Komal',[4,6,7]:'Sampreet'}

Q51.	T20Cricketer={'Name':'Virat', 'Runs':2794, 'Age':31} Barun is trying to delete all the key value pairs of the dictionary using various methods. Which of the following statement will not full fill his wish? a) T20Cricketer.clear() b) del T20Cricketer[‘Name’], T20Cricketer[‘Runs’], T20Cricketer[‘Age’] c) T20Cricketer=dict() d) del T20Cricketer
Ans	del T20Cricketer
Q52.	Predict the output of the following code: T20Cricketer={'Name':'Virat', 'Runs':2794, 'Age':31} print('Virat' in T20Cricketer) a) False b) True c) Error d) ‘Virat’
Ans	a) False
Q53.	Predict the output of the following code: Marks={‘Amar’:87,’Neel’:45,’Rupsa’:92} print(len(Marks)) a) 6 b) 3 c) 5 d) Error
Ans	b) 3
Q54.	Predict the output of the following code: Marks={‘Amar’:87,’Neel’:45,’Rupsa’:92} for i in Marks: print(i, end=’ ‘) a) ‘Amar’ ‘Neel’ ‘Rupsa’ b) 87 45 92 c) 87 45 92 d) ‘Amar’ ‘Neel’ ‘Rupsa’
Ans	a) ‘Amar’ ‘Neel’ ‘Rupsa’
Q55.	Predict the output of the following code: T20Cricketer={'Name':'Virat', 'Runs':2794, 'Age':31} T20Cricketer2={'Name': 'Rohit', 'Age': 33, 'Country': 'India'} T20Cricketer.update(T20Cricketer2) print(T20Cricketer) print(T20Cricketer2) a) {'Name': 'Rohit', 'Runs': 2794, 'Age': 33} {'Name': 'Rohit', 'Age': 33, 'Country': 'India'} b) {'Name': 'Virat', 'Runs': 2794, 'Age': 31, 'Country': 'India'} {'Name': 'Rohit', 'Age': 33, 'Country': 'India'} c) {'Name': 'Rohit', 'Runs': 2794, 'Age': 33, 'Country': 'India'} {'Name': 'Rohit', 'Age': 33, 'Country': 'India'} d) {'Name': 'Rohit', 'Age': 33, 'Country': 'India'} {'Name': 'Rohit', 'Runs': 2794, 'Age': 33, 'Country': 'India'}
Ans	c) {'Name': 'Rohit', 'Runs': 2794, 'Age': 33, 'Country': 'India'} {'Name': 'Rohit', 'Age': 33, 'Country': 'India'}

20 Assertion and reason Based question (1 Mark)	
Mark the correct choice as (a) Both A and R are true and R is the correct explanation for A (b) Both A and R are true and R is not the correct explanation for A (c) A is True but R is False (d) A is false but R is True	
Q1.	Assertion (A): We cannot change the value of an integer variable. Reasoning (R)**:** Integer is immutable.
Ans	(a)
Q2.	Assertion (A): Comments provide extra information in a program. Reasoning (R)**:** Comments are not executed.
Ans	(b)
Q3.	Assertion (A): Strings can be multi-line or single line. Reasoning (R)**:** Strings are mutable.
Ans	(c)
Q4.	Assertion (A): Interactive mode can be used for testing small lines of code. Reasoning (R)**:** It executes the lines of code in an interactive manner.
Ans	(a)
Q5.	Assertion (A): Bug is any error in a program. Reasoning (R)**:** Debugging is the process of removal of error in a program.
Ans	(b)
Q6.	**Assertion.(A)** Python's pass statement is an empty statement **Reason(R)**. An empty statement does nothing
Ans	(a)
Q7.	**Assertion. (A)** The flow of control in a program can occur sequentially, selectively or iteratively. **Reason. (R)**. The sequence construct means that the statement will get executed sequentially.
Ans	(b)
Q8.	**Assertion. (A)** Python statement 'if' represents selection construct. **Reason. (R)**. The selection construct means the execution of a set of statements, depending upon the outcome of a condition.
Ans	(a)
Q9.	**Assertion. (A)** The for loop is a counting loop that works with sequences of values. **Reason. (R)**. The range( ) function generates a sequence of list type.
Ans	(b)
Q10.	**Assertion. (A)** Both break and continue are jump statement **Reason. (R)**. Both break and continue can stop the loops and hence can substitute one another.
Ans	(c)

Q11.	Assertion (A): List can be changed after creation. Reason (R): List are mutable.
Ans	Option a.
Q12.	Assertion (A): remove( ) method removes all elements from a list Reason (R): len ( ) function is used to find the length of list
Ans	d. A is false but R is true.
Q13.	Assertion (A): Elements of a list are separated by comma. Reason (R): List is enclosed by a pair of straight brackets.
Ans	b. Both A and R are true but R is the not correct explanation of A.
Q14.	Assertion (A): clear( ) method removes all elements from a list Reason (R): sort ( ) function is used sort a list in descending order
Ans	c. A is true but R is false.
Q15.	Assertion (A): append( ) method is used to add an element at the end of a list Reason (R): extend ( ) function is used to merge two lists into a single list
Ans	b. Both A and R are true but R is the not correct explanation of A.
Q16.	Assertion (A): Dictionaries are mutable data type. Reasoning (R): We can change the values of the dictionaries.
Ans	(a) Both (A) and (R) are true and (R) is the correct explanation for (A).
Q17.	Assertion (A): Dictionaries are mutable data type. Reasoning (R): We cannot change the keys of the dictionaries.
Ans	(b) Both (A) and (R) are true and (R) is not the correct explanation for (A).
Q18.	Assertion (A):Items in dictionaries are unordered. Reasoning (R):Internally, the key: value pairs of a dictionary are associated with one another with some internal function (called hash-function). This way of linking is called mapping.
Ans	(a) Both (A) and (R) are true and (R) is the correct explanation for (A).
Q19.	Assertion (A): We can update values of a dictionary by the help of keys. Reasoning (R):It is not necessary that the key has to present in the dictionary.
Ans	(c) (A) is true but (R) is false.
Q20.	Assertion (A): We can add new key, value pairsto a dictionary. Reasoning (R): Key to be added must not exist in dictionary and must be unique. If the Key already exist, then it will change the value of existing key and no new entry will be added to dictionary.
Ans.	(a) Both (A) and (R) are true and (R) is the correct explanation for (A).
20 Short Knowledge/Understanding/Application Based Questions (2 Marks)	
Q1.	Write any four rules for naming an identifier.
Ans	(a) It should not be a reserved word (b) It can start with an alphabet or underscore. (c) It cannot contain any special character (d) It can contain digits but not in the starting position.

Q2. State any two differences between '=' and '= ='.

Ans

=	= =
Assignment operator	Relational operator
No return value.	Returns True / False.

Q3. State any two types of operators with example.

Ans Logical operator – and, or, not

Arithmetic operator - +, -, /, *

Q4. What is the difference between '/' and '//'.

Ans

/	//
Division operator	Floor division operator
E.g. 5 / 2 = 2.5	E.g. 5 // 2 = 2

Q5. Give two ways of writing multi-line strings.

Ans

Method 1	Method 2
s= "Hello \ Everyone"	s= ' ' ' Hello Everyone ' ' '

Q6. What is range() function? Give an example.

Ans The range() function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and stops before a specified number.

Syntax: range(start, stop, step)

Example:

```
x = range(3, 7)
for n in x:
  print(n)
```

Output:

3
4
5
6

Q7. What is the difference between break and continue

Ans

Basis for comparison	**break**	**continue**
Use	It is used for the termination of all the remaining iterations of the loop.	It is used for the termination of the only current iteration of the loop.
Control after using break/continue statement	The line which is just after the loop will gain control of the program.	The control will pass to the next iteration of that current loop by skipping the current iteration.
Causes	It performs the termination of the loop.	It performs early execution of the next loop by skipping the current one.

Q8. Write the output of the following code:

```
x=5
while(x<15):
      print(x**2)
      x+=3
```

Ans

```
25
64
121
196
```

Q9. Write the output of the following code:

```
val = 10
total = 0

for count in range(1,val,3):
   total = total + count
   if count % 2 == 0:
      print(count*10)
   else:
      print(count)
print (total)
```

Ans

```
1
40
7
12
```

Q10. Find errors in the following code and write the correct code after underlining it.

```
x = int(input(“Enter value”)
for k in range[0,20]
   if x=k
      print(x+k)
  else:
     Print(x-k)
```

Ans

```
x = int(input("Enter value"))
for k in range(0,20):
   if x==k:
     print(x+k)
   else:
     print(x-k)
```

<table>
<tr><td>Q11.</td><td>Predict the output:-
L1, L2=[1,2,3],[1,2,3]
L3=[1,[2],3]
print(L1==L2)
print(L2==L3)</td></tr>
<tr><td>Ans</td><td>True
False</td></tr>
<tr><td>Q12.</td><td>What is the difference between pop(index) and pop( ) function?</td></tr>
<tr><td>Ans</td><td>pop(index) function deletes the element from i th index of the list.
pop( ) function deletes the last element from the list.</td></tr>
<tr><td>Q13.</td><td>What is the difference between remove( ) and pop( ) function?</td></tr>
<tr><td>Ans</td><td>The argument of pop( ) function is an index. It deletes the element from the given index of the list.
The argument of delete( ) function is an element. It deletes the first occurrence element from the list.</td></tr>
<tr><td>Q14.</td><td>Predict the output of the following code fragment:-
values =[ ]
for i in range (1,4):
values.append(i)
print(values)</td></tr>
<tr><td>Ans</td><td>[1,2,3]</td></tr>
<tr><td>Q15.</td><td>Predict the output of the following code:-
a=[4,3,2,5,6]
print(a[:-3:-1])
print(a[-3:4])</td></tr>
<tr><td>Ans</td><td>[6, 5]
[2, 5]</td></tr>
<tr><td>Q16.</td><td>Write a python statement to create a dictionary ‘Marks5Subs’ having following items: (Please don’t consider the column headers)
<table>
<tr><td>Name</td><td>Marks of 5 Subjects</td></tr>
<tr><td>Sawan</td><td>67,74,56,48,87</td></tr>
<tr><td>Ankit</td><td>34,46,39,21,41</td></tr>
<tr><td>Puja</td><td>91,87,73,82,95</td></tr>
<tr><td>Arnab</td><td>78,98,97,95,99</td></tr>
</table></td></tr>
<tr><td>Ans</td><td>Marks5Subs={'Sawan':[67,74,56,48,87],'Ankit':[34,46,39,21,41],
'Puja':[91,87,73,82,95],'Arnab':[78,98,97,95,99]}</td></tr>
<tr><td>Q17.</td><td>Write a python statement to create a dictionary ‘Currency’ having following items: (Please don’t consider the column headers)
<table>
<tr><td>Country</td><td>Currency</td></tr>
<tr><td>India</td><td>Indian Rupee</td></tr>
<tr><td>Russia</td><td>Ruble</td></tr>
<tr><td>USA</td><td>Dollar</td></tr>
<tr><td>Japan</td><td>Yen</td></tr>
</table></td></tr>
<tr><td>Ans</td><td>Currency={'India':'Indian Rupee','Russia':'Ruble',
'USA':'Dollar','Japan':'Yen'}</td></tr>
</table>

<table>
<tr><td>Q18.</td><td>Write a python statement to create a dictionary ‘NationalBird’ having following items: (Please don’t consider the column headers)
<table>
<tr><td>Country</td><td>National Bird</td></tr>
<tr><td>India</td><td>Peacock</td></tr>
<tr><td>Australia</td><td>Emu</td></tr>
<tr><td>Bahamas</td><td>Flamingo</td></tr>
<tr><td>Italy</td><td>Sparrow</td></tr>
<tr><td>New Zealand</td><td>Kiwi</td></tr>
</table></td></tr>
<tr><td>Ans</td><td>NationalBird={'India':'Peacock','Australia':'Emu','Bahamas':'Flamingo',
'Italy':'Sparrow','New Zealand':'Kiwi'}</td></tr>
<tr><td>Q19.</td><td>Ranit has written a python code to create a dictionary and display the values of the dictionary using a loop but it is showing errors. Help him to find out errors and underline the corrections.
Stds={'IX';153,'X';143,'XI';147,'XII';89}
for i in Std:
Print(i)</td></tr>
<tr><td>Ans</td><td>Stds={'IX':153,'X':143,'XI':147,'XII':89}
for i in <u>Stds</u>:
print(<u>Stds[i]</u>)</td></tr>
<tr><td>Q20.</td><td>Sharmili has written a python code to create a dictionary Stds that stores student’s strength of classes IX, X, XI and XII and it will calculate the total student’s strength it is showing errors, Help her to find out errors and underline the corrections.
Stds={'IX':153,'X':143,'XI':147,'XII':89
Sum=1
for i in Stds.value():
Sum=+i
print(Sum)</td></tr>
<tr><td>Ans.</td><td>Stds={'IX':153,'X':143,'XI':147,'XII':89<u>}</u>
<u>Sum=0</u>
for i in Stds.<u>values</u>():
Sum<u>+=</u>i
print(Sum)</td></tr>
<tr><td colspan="2">20 Short Knowledge/Understanding/Application Based Questions (3 Marks)</td></tr>
<tr><td>Q1.</td><td>What is data type conversion? State its two types with example.</td></tr>
<tr><td>Ans</td><td>Conversion of one data type to another is known as data type conversion.
Two types - implicit and explicit(type casting).
Implicit - 5 / 2 = 2.5
Explicit - int(3.25) → 3</td></tr>
<tr><td>Q2.</td><td>Define Dynamic Typing. Give an example.</td></tr>
<tr><td>Ans</td><td>The value allotted to a variable can be changed dynamically in a program.
E.g.:
a=10
print(a)
a=”Hello” # changing the value of the variable</td></tr>
</table>

Q3.	Using example, explain the difference between mutable and immutable datatype.
Ans	If the value of variable of a data type can be changed without affecting its address then it is known as mutable data type, else it is known as immutable data type.
Q4.	What is the use of comments in a program ? What are its two types ?
Ans	Comments provide extra information for increasing the readability of a program. Two types - single line comment and multi-line comment.
Q5.	What do you mean by the precedence of operators ?
Ans	Operator precedence affects how an expression is evaluated. For example, x = 7 + 3 * 2; here, x is assigned 13, not 20 because operator * has higher precedence than +, so it first multiplies 3*2 and then adds into 7.
Q6.	Write a program to find the sum of the given series. 1 + x^2/2 + x^3/3 + … x^n/n
Ans	n=int(input("Enter the number of terms:")) x=int(input("Enter the value of x:")) sum1=1 for i in range(2,n+1): sum1=sum1+((x**i)/i) print("The sum of series is",round(sum1,2))
Q7.	Write a program to reverse a number using while loop.

Ans	num = int (input (“Enter a number: “))
	reversed_num = 0
	while num != 0:
	  digit = num % 10
	  reversed_num = reversed_num * 10 + digit
	  num = num/10
	print("Reversed Number: " + str(reversed_num))

Q8.	Write a program to calculate the factorial of a number.

Ans	n = int (input (“Enter a number: “))
	factorial = 1
	if n >= 1:
	  for i in range (1, n+1):
	  factorial = factorial *i
	print (“Factorial of the given number is: “, factorial)

Q9.	Write a program to check whether the number entered by the user is Perfect number.

Ans	n = int(input("Enter any number: "))
	sum1 = 0
	for i in range(1, n):
	  if(n % i == 0):
	  sum1 = sum1 + i
	if (sum1 == n):
	  print("The number is a Perfect number!")
	else:
	  print("The number is not a Perfect number!")

Q10.	Write a program to print the given pattern. * * * * * * * * * * * * * * *

Ans	for i in range(1, 6):
	  for j in range(1, i+1):
	  print('*', end = " ")
	  print()

Q11.	Write a program to find the average from a given list of integers.
Ans	L=[5,8,3,4,6] sum=0 for i in L: sum+=i avg=sum/len(L) print(avg)

Q12.	Predict the output of the following code:-
	M=[]
	M1=[]
	M2=[]
	for i in range(1,10):
	  M.append(i)
	for i in range(10,1,–2):
	  M1.append(i)
	for i in range(len(M1)):
	  M2.append(M1[i]+M[i])
	M2.append(len(M)-len(M1))
	  print(M2)

Ans	[11, 10, 9, 8, 7, 4]
Q13.	What is the difference between sort( ) and sorted( ) function?
Ans	sort( ) function will modify the list from which the function has been called while sorted( ) function will create a new list which is given as argument. sort( ) function works upon list onlt while sorted( ) function will work upon any iterative sequence.

Q14.	Predict output of the following code:-
	for Name in ['Jayes', 'Ramya', 'Taruna', 'Suraj'] :
	  print(Name)
	  if Name[0]=='T':
	  break
	  else :
	  print('Finished!')
	  print('Got it!')

Ans	Jayes Finished! Got it! Ramya Finished! Got it! Taruna

Q15.	Write a program that takes first 5 and last 5 elements of a list and stores them into another list.

Ans	L=[1,1,2,3,5,4,7,9,5,4,9,6]
	if len(L)<10:
	  print("Insufficient elements")
	else:
	  LST=L[:5]+L[-5:]
	  print(L)
	  print(LST)

Q16.	Consider the following Dictionary. Capital={'India':'New Delhi', 'Iran':'Teheran', 'Nepal':'Kathmandu', 'Russia':'Moscow'} Write statements to do the following: a) To insert a new item for the country Japan. b) To display name of the countries from the dictionary Capital. c) To display name of the capitals from the dictionary Capital.
Ans	a) Capital['Japan']='Tokyo' b) print(Capital.keys()) c) print(Capital.values())
Q17.	Consider the following Dictionary. Goals2023={‘Messi’:26,’Ronaldo’:35,‘Haaland’:25,’Neymar’:20} Write statements to do the following: a) To modify the Goals of Neymar as 18. b) To delete the record of Haaland. c) To display the goals of Ronaldo.
Ans	a) Goals2023['Neymar']=18 b) del Goals2023['Haaland'] c) print(Goals2023[‘Ronaldo’])
Q18.	Write a python program to store details of five teachers having Employee ID, Name and Designation to a dictionary and display only the details of those teachers whose name starts with ‘R’ and designation is ‘PGT’.

Ans	Teacher={'Emp1':{'EmpId':3698,'Name':'Rabi','Desig':'PGT'},
	'Emp2':{'EmpId':9821,'Name':'Sachin','Desig':'PRT'},
	'Emp3':{'EmpId':8219,'Name':'Ruksana','Desig':'TGT'},
	'Emp4':{'EmpId':2195,'Name':'Martin','Desig':'TGT'},
	'Emp5':{'EmpId':1975,'Name':'Robin','Desig':'PGT'}}
	for i in Teacher.values():
	  if i['Name'][0]=='R' and i['Desig']=='PGT':
	  print(i)

Q19.	Predict the output of the following code:
	Goals2023={'Messi':26,'Ronaldo':35,'Haaland':25,'Neymar':20}
	for i in Goals2023:
	  if len(i)>6:
	  print(i)
	for i in Goals2023:
	  if Goals2023[i]>25:
	  print(i,Goals2023[i])
	for i in Goals2023:
	  if Goals2023[i]%2==0:
	  print(i,Goals2023[i])

Ans	Ronaldo Haaland Messi 26 Ronaldo 35 Messi 26 Neymar 20

Q20.	Soham wants to write a Python Code to calculate frequency of each distinct element of a list but he is struggling at some points help him to complete the code. Example: Input: [12, 34, 21, 45, 21, 45, 12, 21, 32, 21, 21] Output: {12: 2, 34: 1, 21: 5, 45: 2, 32: 1} Code: L=[12,34,21,45,21,45,12,21,32,21,21] D={} for _____: #Statement 1 if i_____D: #Statement 2 _____ #Statement 3 else: D[i]+=1 print(L) print(D) a) Complete Statement 1 to traverse each element of the list one by one. b) Complete Statement 2 to check whether i is not present in D as keys. c) Complete Statement 3 to insert an entry for i in D with appropriate value.
Ans.	a) Statement 1: for i in L: b) Statement 2: if i not in D: c) Statement 3: D[i]=1
20 Short Knowledge/Understanding/Application Based Questions (4 Marks)	

Q1.	Vedansh is a Python programmer working in a school. He has written the following code, but it contains mistakes.
	
	n1=Int (input ("Enter the number)) # statement 1
	2n=int (input ("Enter the number")) # statement 2
	rem= n1 % n2 # statement 3
	print("rem") # statement 4
	As a Python expert, help him by answering the following questions:
	(a) Identify the statements that don't contain any errors.
	(b) Write the correct code for the statements containing error.
	(c) Which type of operator is being used in statement 3.
	(d) What will be the output if the input is 10 and 6 respectively for n1 and n2.

Ans	(a) statement 3
	(b) n1=int (input ("Enter the number")) # statement 1
	n2=int (input ("Enter the number")) # statement 2
	print(rem) # statement 4
	(c) Arithmetic operator
	(d) 4

<table>
<tr><td>Q2.</td><td>Soham has chosen the following names for some variables, give reasons why they are invalid :
(a) 1sum
(b) sum@
(c) sum of num
(d) class</td></tr>
<tr><td>Ans</td><td>(e) 1sum - starts with digit
(f) sum@ - contains special character ‘@’
(g) sum of num - contains space
(h) class - keyword</td></tr>
<tr><td>Q3.</td><td>Rohan is trying to guess the size of following strings , help him to do so:
<table><tr><td>(a)
‘\n’</td><td>(b)
“Ram’s”</td></tr><tr><td>(c)
“ “ “ Hi
All “ “ “</td><td>(d)
“Hi \
All”</td></tr></table></td></tr>
<tr><td>Ans</td><td>(a) 1 (b) 5 (c) 6 (d) 5</td></tr>
<tr><td>Q4.</td><td>Find the output of the following code:
a,b,c=10,20,30
a,c,b=b-5,a-3,c-6
print(a,b,c)</td></tr>
<tr><td>Ans</td><td>15 24 7</td></tr>
<tr><td>Q5.</td><td>Find the output of the following code snippets:
(a) type(‘None’) (b) type(None)
(c) print(print(“OK”)) (d) type(0o56)</td></tr>
<tr><td>Ans</td><td>(a) string (b) None (c) None (d) int</td></tr>
<tr><td>Q6.</td><td>What do you mean by looping construct in Python? Explain for loop and while loop with their syntax and appropriate examples.</td></tr>
<tr><td>Ans</td><td>The looping construct means repetition of a set of statements on the basis of a condition test. Furthermore, till the time a condition turns out to be true or false depending upon the loop, the repetition of a set of statements takes place again and again.
for loop
A for loop is a type of loop that runs for a preset number of times. It also has the ability to iterate over the items of any sequence, such as a list or a string.</td></tr>
</table>

	Syntax
	for i in <collection>:
	<loop body>
	Example
	for i in range(10): # collection of numbers from 0 to 9
	print(i)
	Here, collection is a list of objects. The loop variable, i, takes on the value of the next element in collection each time through the loop. The code within loop body keeps on running until i reach the end of the collection.
	while loop
	With the while loop, we can execute a block of code as long as a condition is true.
	Syntax
	while <condition>:
	<loop body>
	In a while loop, the condition is first checked. If it is true , the code in loop body is executed. This process will repeat until the condition becomes false.
	This piece of code prints out integers between 0 and 9 .
	Example
	n = 0
	while n < 10: # while n is less than 10,
	print(n) # print out the value of n
	n += 1 #

Q7.	What do you mean by jumping statements in Python? Explain break, continue and pass with appropriate examples.

Ans	In Python, jumping statements are used to control the flow of a program by altering the normal execution sequence. They allow you to change the order in which statements are executed in a loop or conditional block. The three common jumping statements in Python are break, continue, and pass.
	break:
	The break statement is used to exit the current loop prematurely, whether it's a for loop or a while loop.
	It is typically used when a certain condition is met, and you want to terminate the loop immediately.
	Example:
	for i in range(1, 6):
	if i == 3:
	break # This will exit the loop when i is equal to 3
	print(i)
	continue:
	The continue statement is used to skip the current iteration of a loop and proceed to the next iteration.
	It is often used when you want to skip some specific values or conditions but continue with the loop.
	Example:
	for i in range(1, 6):
	if i == 3:
	continue # This will skip iteration when i is equal to 3
	print(i)
	pass:
	
	The pass statement is a placeholder statement that does nothing. It is often used as a placeholder when you need a statement syntactically but don't want to execute any code.

	Example:
	for i in range(1, 4):
	  if i == 2:
	  pass # This will do nothing when i is equal to 2
	  else:
	  print(i)

Q8.	Write a program to check if input number is a prime number.

Ans	num = int(input("Enter a number: "))
	if num > 1:
	  # check for factors
	  for i in range(2,num):
	  if (num % i) == 0:
	  print(num,"is not a prime number")
	  break
	  else:
	  print(num,"is a prime number")
	# if input number is less than
	# or equal to 1, it is not prime
	else:
	  print(num,"is not a prime number")

Q9.	Write a program to check whether a year is leap year or not.

Ans	input_year = int(input("Enter the Year to be checked: "))
	if ((input_year%400 == 0) or ((input_year%4 == 0) and (input_year%100 != 0))):
	  print("%d is Leap Year" %input_year)
	else:
	  print("%d is Not the Leap Year" %input_year)

Q10.	Write a program to display the Fibonacci series upto nth term.

Ans	#Python program to generate Fibonacci series until 'n' value
	n = int(input("Enter the value of 'n': "))
	a = 0
	b = 1
	sum = 0
	count = 1
	print("Fibonacci Series: ", end = " ")
	while(count <= n):
	  print(sum, end = " ")
	  count += 1
	  a = b
	  b = sum
	  sum = a + b

Q11.	Predict the output of the following code:-
	  Moves=[11, 22, 33, 44]
	  Queen=Moves
	  Moves[2]+=22
	  L=len(Moves)
	  for i in range (L):
	  print (“Now@”, Queen[L-i-1], “#”, Moves [i])

Ans	Now@ 44 # 11 Now@ 55 # 22 Now@ 22 # 55 Now@ 11 # 44

Q12.	Write a program to count positive numbers, negetive numbers and zeroes from a list of integers. The list elements will be entered by the user.

Ans	L=[]
	p,n,z=0,0,0
	s=int(input("Enter size of the list:"))
	for i in range(s):
	  x=int(input("Enter element:"))
	  L.append(x)
	  if x>0:
	  p=p+1
	  elif x<0:
	  n=n+1
	  else:
	  z=z+1
	print(L)
	print("Postive Numbers=",p)
	print("Negative Numbers=",n)
	print("Zeroes=",z)

Q13.	Write a program to input a list and an element and remove all occurences of the given element from the list.

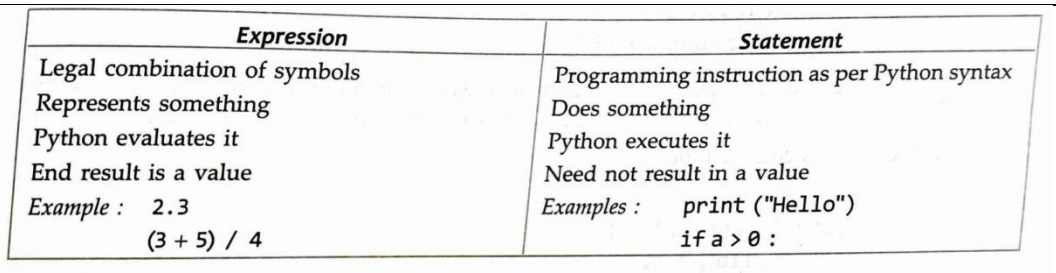
Ans	Lst=eval(input("Enter a list"))
	item=int(input("Enter thr item to remove"))
	c=Lst.count(item)
	if c==0:
	  print("Item not found")
	else:
	  while(c>0):
	  i=Lst.index(item)
	  Lst.pop(i)
	  c-=1
	print(Lst)

Q14.	Given a list of integers, write a program to sum of even mumbers and odd numbers.

Ans	L=[5,8,9,7,5,4]
	sumeven, sumodd=0,0
	for i in L:
	  if i%2==0:
	  sumeven+=i
	  else:
	  sumodd+=i
	print(L)
	print("Sum of even Numbers=",sumeven)
	print("Sum of odd Numbers=",sumodd)

Q15.	Identify the operators along with their names from the following statements:- a. 5 in [1,2,3,4,5] b. [1,2,3]*2 c. [1,2,3,4,5][1:3] d. [1,2,3]+[4,5]
Ans	a. in – membership operator b. * - replication operator c. : - slice operator d. + - concatenation operator

Q16.	Predict the output of the following code: Teacher1,Teacher2={'EmpId':3698,'Name':'Robin','Desig':'PGT', 'Sub':'Chemistry'},{'EmpId':9821, 'Name':'Sachin','Desig':'PRT','HomeTown':'Patna'} print(Teacher1) print(Teacher2) Teacher1.update(Teacher2) print(Teacher1) print(Teacher2)
Ans	{'EmpId': 3698, 'Name': 'Robin', 'Desig': 'PGT', 'Sub': 'Chemistry'} {'EmpId': 9821, 'Name': 'Sachin', 'Desig': 'PRT', 'HomeTown': 'Patna'} {'EmpId': 9821, 'Name': 'Sachin', 'Desig': 'PRT', 'Sub': 'Chemistry', 'HomeTown': 'Patna'} {'EmpId': 9821, 'Name': 'Sachin', 'Desig': 'PRT', 'HomeTown': 'Patna'}
Q17.	Predict the output of the following code: Marks5Subs={'Sawan':[67,74,56,48,87],'Ankit':[34,46,39,21,41], 'Puja':[91,87,73,82,95],'Arnab':[78,98,97,95,99]} print(max(Marks5Subs['Sawan'])) print(min(Marks5Subs['Ankit'])) print(len(Marks5Subs['Arnab'])) print(len(Marks5Subs))
Ans	87 21 5 4
Q18.	Predict the output of the following code: Age={'Sawan':67,'Ankit':34,'Puja':21,'Arnab':23} print(list(Age.items())) del Age['Sawan'] print(Age) Age.clear() print(Age) del Age print(Age)
Ans	[('Sawan', 67), ('Ankit', 34), ('Puja', 21), ('Arnab', 23)] {'Ankit': 34, 'Puja': 21, 'Arnab': 23} {} NameError: name 'Age' is not defined
Q19.	Predict the output of the following code: Result={'PT1':{'Suresh':35,'Kabir':29,'Lisa':17,'Hina':36}, 'HYE':{'Suresh':87,'Kabir':56,'Lisa':87,'Hina':65}, 'PT2':{'Suresh':37,'Kabir':23,'Lisa':27,'Hina':33}, 'SEE':{'Suresh':78,'Kabir':65,'Lisa':89,'Hina':75}} T_Suresh=T_Kabir=T_Lisa=T_Hina=0 for i in Result: T_Suresh+=Result[i]['Suresh'] T_Kabir+=Result[i]['Kabir'] T_Lisa+=Result[i]['Lisa'] T_Hina+=Result[i]['Hina'] print(T_Suresh)

	print(T_Kabir) print(T_Lisa) print(T_Hina)
Ans	237 173 220 209
Q20.	Predict the output of the following code: Teacher1=Teacher2={'EmpId':3698,'Name':'Robin','Desig':'TGT'} Teacher1['Desig']='PGT' Teacher2['Sub']='CS' print(Teacher1) print(id(Teacher1)) print(Teacher2) print(id(Teacher2))
Ans.	{'EmpId': 3698, 'Name': 'Robin', 'Desig': 'PGT', 'Sub': 'CS'} 50519352 {'EmpId': 3698, 'Name': 'Robin', 'Desig': 'PGT', 'Sub': 'CS'} 50519352
15 Case Based Questions (5 Marks)	
Q1.	Namita is trying to understand the concept of literal, help him by answering the following questions: (a) what is literal (b) state any two types of literals (c) Name the special literal (d) give an example of integer literal (e) what is Boolean literal ?
Ans	(a) Literal represents a value of a particular data type (b) string literal and boolean literal. (c) None (d) 527 (e) True / False
Q2.	Rakesh is unable to understand the difference between statement and expression, write differences between them along with examples.
Ans	

Expression	Statement
Legal combination of symbols	Programming instruction as per Python syntax
Represents something	Does something
Python evaluates it	Python executes it
End result is a value	Need not result in a value
Example : `2.3` `(3 + 5) / 4`	*Examples :* `print ("Hello")` `if a > 0 :`

Q3.	Shilu has to write a program to accept the name of a person and greet him/her in the following manner: Hello <name>, welcome to our class. Help Shilu to write the program.
Ans	n=input("Enter the name:") print("Hello",n, "welcome to our class")
Q4.	Write a program to accept two numbers and print their sum in the following manner: The sum of <n1> and <n2> is <n1+n2>.
Ans	n1=int(input("Enter the number:")) n2=int(input("Enter the number:")) print("The sum of" ,n1," and ", n2," is",n1+n2)
Q5.	Find the output of the following: (a) " " and "Hello" (b) 2 and 4 (c) 'a' or 'b' (d) True and 'Hi' (e) 0 or 5
Ans	(f) " " and "Hello" → "Hello (g) 2 and 4 → 4 (h) 'a' or 'b' → a (i) True and 'Hi' → Hi (j) 0 or 5 - 5
Q6.	Mr. Aakash wants to calculate electricity charges based on the number of consumed electricity units and other charges. Write a program in Python to generate electricity bill as per the following conditions. i. If unit consumed <= 100 then cost per unit is Rs 3.46 ii. If unit consumed >= 101 and <= 300 then cost per unit is Rs 7.43 iii. If unit consumed >= 301 and <= 500 then cost per unit is Rs 10.32 iv. If unit consumed >= 501 then the cost per unit is Rs 11.71 v. Line rent is Rs 1.45 per unit. vi. Additional fixed Meter rent is Rs 100. vii. The tax on the bill is 16 percent which can be taken as 0.16.

Ans	unit = int(input("Enter your unit: "))
	if unit <= 100:
	bill = unit * 3.46
	elif unit >= 101 and unit <= 300:
	bill = 346 + ((unit - 100) * 7.43)
	elif unit >= 301 and unit <= 500:

```
    bill = 346 + 1486 + ((unit - 300) * 10.32)
else:
    bill = 346 + 1486 + 2064 + ((unit - 500) * 11.71)
print("Bill Per Unit:",bill)
bill = bill + (unit*1.45)
print("Bill after adding Line rent:",bill)
bill = bill + 100
print("Bill after adding Meter rent:",bill)
bill = bill + (bill*0.16)
print("Total Bill after adding tax:",bill)
```

Q7. Mr. Ravi is a class teacher in Modern Public School. He wants to determine the student's grade based on the results of five subjects and the criteria given below.

Average Mark	**Grade**
91-100	A1
81-90	A2
71-80	B1
61-70	B2
51-60	C1
41-50	C2
33-40	D
21-32	E1
0-20	E2

Write an appropriate program in Python to find out the grade of a student.

Ans

```
print("Enter Marks Obtained in 5 Subjects: ")
markOne = int(input())
markTwo = int(input())
markThree = int(input())
markFour = int(input())
markFive = int(input())
tot = markOne+markTwo+markThree+markFour+markFive
avg = tot/5
if avg>=91 and avg<=100:
    print("Your Grade is A1")
elif avg>=81 and avg<91:
    print("Your Grade is A2")
elif avg>=71 and avg<81:
    print("Your Grade is B1")
elif avg>=61 and avg<71:
    print("Your Grade is B2")
elif avg>=51 and avg<61:
    print("Your Grade is C1")
elif avg>=41 and avg<51:
    print("Your Grade is C2")
elif avg>=33 and avg<41:
    print("Your Grade is D")
```

	elif avg>=21 and avg<33:
	print("Your Grade is E1")
	elif avg>=0 and avg<21:
	print("Your Grade is E2")
	else:
	print("Invalid Input!")

Q8.	A list is a standard data type in Python that can store a sequence of values belonging to any type. Lists are enclosed in a pair of square brackets. These are mutable, i.e, elements can be changed by the user. Every element of a list has an index. Indexing begins from zero. Questions:- I.List defined within a list is called:- a. nested list b. super list c. sub list d. hidden list II. In Python, list is of type:- a. Immutable b. Mutable c. Both a & B d. None of a & b III. If a list contains n elements, then the index of the last element will be:- a. 0 b. n c. n+1 d. n-1 IV. Which type of the bracket is used to define a list? a. ( ) b. { } c. [ ] d. <> V. List can contain values of these types:- a. integers b. float c. string d. all of these
Ans	I. a II.b III.d IV.c V.d
Q9.	Amit has created two lists L1=[6,2,3,8] and L2=[1,5,4] He has been asked by his teacher to write the code for the following tasks:- I. To predict the output of the code:- L3=L2.extend(L1) print(L3) II. To display smallest number from L3 III.To add 2nd element from L1 and 3rd element from L2 IV. To arrange the elements of L3 in descending order V. To predict the output : L1[:2]+L2[2:]
Ans	I. [1,5,4,6,2,3,8] II. min(L3) III. L1[1] + L2[2] IV. L3.sort(reverse=True) V. [6,2,4]
Q10.	Rakesh wants to write a program to count the number of vowels from the word 'Alexander' by converting it into a list. But the program does not run due to errors. Help Rakesh to identify and rectify the errors so that program can run:- L=List('Alexander') count==0 For i in L: if i within 'aeiouAEIOU' count=+1 print(count)
Ans	L=list('Alexander') count=0 for i in L: if i in 'aeiouAEIOU' count+=1 print(count)

Q11.	Raman has stored record of a student in a list as follows:- rec=['Thomas','C-25',[56,98,99,72,69],78.8] Suggest him the Python statements to do the following tasks:- a. To find the percentage b. To find marks of 5th subject c. Maximum marks of the student d. To find total marks e. To change the name from 'Thomas' to ' Charles'
Ans	a. rec[3] b. rec[2][4] c. max(rec[2]) d. rec[0]+rec[1]+rec[2]+rec[3]+rec[4] or sum(rec) e. rec[0]='Charles'

Q12.	Rehana has a list of both positive numebers. She has been given a task to separate positive and negetaive numbers into two different lists and finally to delete the original list. She has written a code where some statements incomplete. Complete the imcomplete statements by filling in the blanks:-
	Numbers=[5,-8,9,-7,5,-4]
	Pos, Neg= ____ #Statement 1: To initialize empty lists
	for i in range( ): # Statement 2: To write the range to access all elements
	if Numbers[i]>=0:
	_____ # Statement 3: To add element in POS
	else:
	_____ # Statement 4: To add element in another list
	_____ #Statement 5: To delete the original list
	print (Pos)
	print(Neg)
	print("Task Completed")

Ans	a. Statement 1: Pos, Neg=[],[] b. Statement 2: for i in range(len(Numbers)): c. Statement 3: Pos.append(Numbers[i]) d. Statement 4: Neg.append(Numbers[i]) e. Statement 5: del Numbers
Q13.	Write a menu driven program to store marks of students with the following features: Press 1 to add a new student's record. Press 2 to update an existing student's record. Press 3 to delete an existing student's record who have taken TC Press 4 to display a particular student's record. Press 5 to display records of all students Press 6 to exit

Ans	Record={}
	while True:
	print('Press 1 to add a new student's record')
	print('Press 2 to update an existing student's record')
	print('Press 3 to delete an existing student's record who have taken TC')
	print('Press 4 to display a particular student's record')
	print('Press 5 to display records of all students')
	print('Press 6 to exit')
	op=int(input('enter the value'))
	if op==1:

	Name=input('Enter Name')
	Marks=int(input('Enter Marks'))
	Record[Name]=Marks
	elif op==2:
	Name=input('Enter Name')
	Marks=int(input('Enter Marks'))
	Record[Name]=Marks
	elif op==3:
	Name=input('Enter Name')
	del Record[Name]
	elif op==4:
	Name=input('Enter Name')
	print(Record[Name])
	elif op==5:
	print(Record)
	elif op==6:
	break
	else:
	print('Wrong Choice')

Q14.	Write a menu driven program to show category wise student enrolment details of a KV with the following features. Press 1 to add a new category. Press 2 to update an existing category. Press 3 to delete an existing category Press 4 to display enrolment of a particular category. Press 5 to display all category wise enrolment. Press 6 to exit

Ans	Enrol={}
	while True:
	print('Press 1 to add a new category')
	print('Press 2 to update an existing category')
	print('Press 3 to delete an existing category')
	print('Press 4 to display enrolment of a particular category')
	print('Press 5 to display all category wise enrolment')
	print('Press 6 to exit')
	op=int(input('enter the value'))
	if op==1:
	Cat=input('Enter Category')
	Tot=int(input('Enter enrolment under the Category'))
	Enrol[Cat]=Tot
	elif op==2:
	Cat=input('Enter Category')
	Tot=int(input('Enter enrolment under the Category'))
	Enrol[Cat]=Tot
	elif op==3:
	Cat=input('Enter Category')
	del Enrol[Cat]
	elif op==4:
	Cat=input('Enter Category')
	print(Enrol[Cat])

```
    elif op==5:
        print(Enrol)
    elif op==6:
        break
    else:
        print('Wrong Choice')
```

Q15. Write a menu driven program to simulate Bank Application with the following features:
Press 1 to open a savings bank account
Press 2 to deposit money
Press 3 to withdraw money
Press 4 to check balance
Press 5 to exit

Ans

```
Acc={}
AccNo=1000
while True:
    print('Press 1 to open a savings bank account')
    print('Press 2 to deposit money')
    print('Press 3 to withdraw money')
    print('Press 4 to check balance')
    print('Press 5 to exit')
    op=int(input('enter the value'))
    if op==1:
        Name=input('Enter Name')
        Age=int(input('Enter Age'))
        AccNo+=1
        Acc[AccNo]={'Name':Name,'Age':Age,'Bal':0}
    elif op==2:
        Ac=int(input('Enter Account No'))
        Amt=int(input('Enter the amount to be deposited'))
        for i in Acc:
            if i==Ac:
                Acc[i]['Bal']+=Amt
    elif op==3:
        Ac=int(input('Enter Account No'))
        Amt=int(input('Enter the amount to be deposited'))
        for i in Acc:
            if i==Ac:
                Acc[i]['Bal']-=Amt
    elif op==4:
        Ac=int(input('Enter Account No'))
        for i in Acc:
            if i==Ac:
                print(Acc[i])
    elif op==5:
        break
    else:
        print('Wrong Choice')
```