

KVS ERNAKULAM REGION



केन्द्रीय विद्यालय संगठन, एर्नाकुलम क्षेत्र

**KENDRIYA VIDYALAYA SANGTAHAN  
ERNAKULAM REGION**

**STUDENT SUPPORT MATERIAL**

TERM-1 - CLASS XII - MCQ - 2021-22

**Computer Science (083)**

Based on latest CBSE Exam Pattern for the Session 2021-22



## CHIEF PATRON



**Mr R Senthil Kumar**  
Deputy Commissioner, KVSRO Ernakulam

## PATRONS



**Mrs. Deepti Nair**  
Asst. Commissioner, KVSRO Ernakulam



**Mr. Santhosh Kumar N**  
Assistant Commissioner, KVSRO Ernakulam

## CO-ORDINATOR

**Mr. Jyothi Mohan N V**  
Principal, Kendriya Vidyalaya S A P, Peroorkada



PREPARED BY PGT COMPUTER SCIENCE



- ✚ **Mrs. RADHA MOHAN** ,PGT CS KV PATTOM SHIFT-1
- ✚ **Mrs. SINDHU S** , PGT CS, KV KOLLAM
- ✚ **Mrs HEMA R**, PGT CS, KV ADOOR
- ✚ **Mrs AMBILY KRISHNAN**, PGT CS, KV CRPF PALLIPURAM
- ✚ **Mrs SUJA P NAIR**, PGT CS ,KV ERNAKULAM
- ✚ **Mrs. SUNI ABHRAHAM**, PGT CS, KV RB KOTTAYAM
- ✚ **Mrs. VRINDA S**, PGT CS , KV PORT TRUST
- ✚ **Mr. SREEJITH K V**, PGT CS, KV PANGODE
- ✚ **Mrs ASWATHI V**, PGT CS, KV CALICUT

आर सेन्धिल कुमार  
उपायुक्त

*R. Senthil Kumar*  
Deputy Commissioner



केन्द्रीय विद्यालय संगठन,  
क्षेत्रीय कार्यालय, एरणाकुलम

**KENDRIYA VIDYALAYA SANGATHAN**  
**REGIONAL OFFICE,**  
**ERNAKULAM,**  
**KOCHI – 682 020**  
Ph. No.0484- 2205111(DC), 2203091(Fax))  
Website: [www.roernakulam.kvs.gov.in](http://www.roernakulam.kvs.gov.in)  
Email : [dcernakulamregion@gmail.com](mailto:dcernakulamregion@gmail.com)

*F.31/Acad/KVS(EKM)*

*Dated: 01.11.2021*

**Message**

*I feel immense pleasure to publish the study material for class **XII Computer Science (083)**. This support material is prepared incorporating all the recent changes in curriculum and assessment process made by CBSE. I am sure it will definitely be of great help to class **XII** students of all Kendriya Vidyalayas.*

*Getting acquainted with the latest changes will help students to prepare well for the board examination and enable students to face case based and Multiple-Choice Questions with confidence. This support material has been prepared by a team of dedicated and veteran teachers with expertise in their respective subjects.*

*The Support material contains all the important aspects required by the students- the design of question paper, term wise split up syllabus, summary of all the chapters, important formulas, Sample question papers, problem solving and Case study questions.*

*I hope that this Support Material will be used by students and teachers as well and will prove to be a good tool for quick revision.*

*I would like to express my sincere gratitude to the In- charge principal and all the teachers who have relentlessly worked for the preparation of this study material. Their enormous contribution in making this project successful is praiseworthy.*

*Meticulous planning blended with hard work, effective time management and sincerity will help the students to reach the pinnacle of success.*

*Wish you all the best*

  
( R Senthil Kumar )

*Sh. Jyothimohan N V*  
Principal  
Kendriya Vidyalaya  
SAP Peroorkada

CBSE Term wise Syllabus

Revision of Python topics covered in Class XI

**1**

Page No. 1 to 64

Functions

**2**

Page No 65 to 92

Introduction to files

**3**

Page No. 93 to 111

Binary file

**4**

Page No.112 to 160

CSV file

**5**

Page No.161 to 178

CBSE Term – 1 Sample Paper with Solution

KVS Ernakulam Region - Practice Test Question  
paper with Marking Scheme

## Computer Science

### CLASS-XII

Code No. 083

2021-22

#### 1. Prerequisites

Computer Science- Class XI

#### 2. Learning Outcomes

Student should be able to

- a) apply the concept of function.
- b) explain and use the concept of file handling.
- c) use basic data structure: Stacks.
- d) explain basics of computer networks.
- e) use Database concepts, SQL along with connectivity between Python and SQL.

#### 3. Distribution of Marks:

Unit No.	Unit Name	Marks	Periods	
			Theory	Practical
I	Computational Thinking and Programming - 2	40	50	25
II	Computer Networks	10	10	---
III	Database Management	20	20	15
	Total	70	80	40

Unit No	Unit Name	Term-1	Term-2
I	Computational Thinking and Programming - 2	35	5
II	Computer Networks	---	10
III	Database Management	---	20
	Total	35	35

## 4. Unit wise Syllabus

### TERM 1:

#### **Unit I: Computational Thinking and Programming – 2**

- 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)
- Introduction to files, types of files (Text file, Binary file, CSV file), relative and absolute paths
- Text file: opening a text file, text file open modes (r, r+, w, w+, a, a+), closing a text file, 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( )

### TERM 2:

#### **Unit I: Computational Thinking and Programming – 2**

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

#### **Unit II: Computer Networks**

- 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
- Introduction to web services: WWW, Hyper Text Markup Language (HTML), Extensible Markup Language (XML), domain names, URL, website, web browser, web servers, web hosting

## Unit III: Database Management

- 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, 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

### 5. Practical

S.No		Marks (Total 30)	Term-1 (15 Marks)	Term-2 (15 Marks)
1	<b>Lab Test:</b>			
	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: Term – 1 : Minimum 15 Python programs based on Term - 1 Syllabus Term – 2 : <ul style="list-style-type: none"> <li>Minimum 3 Python programs based on Term-2 Syllabus</li> <li>SQL Queries – Minimum 5 sets using one table / two tables.</li> <li>Minimum 2 programs based on Python - SQL connectivity.</li> </ul>	7	4	3
3	Project (using concepts learnt in Classes 11 and 12) Term – 1 : Synopsis of the project to be submitted by the students (documentation only, may not submit the code during Term - 1)  Term - 2 : Final coding + Viva voce (Student will be allowed to modify their Term 1 document and submit the final executable code.)	8	3	5
4	Viva voce	3	2	1

## **6. Suggested Practical List:**

### **Term-1**

#### **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.

### **Term-2**

#### **Python Programming**

- 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.

#### **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
- Similar exercise may be framed for other cases.
- Integrate SQL with Python by importing suitable module.



## **7. Suggested Reading Material**

- NCERT Textbook for COMPUTER SCIENCE (Class XII)
- Support Materials on the CBSE website.

## **8. Project**

The aim of the class project is to create something that is tangible and useful using Python file handling/ Python-SQL connectivity. This should be done in groups of two to three students and should be started by students at least 6 months before the submission deadline. The aim here is to find a real world problem that is worthwhile to solve.

Students are encouraged to visit local businesses and ask them about the problems that they are facing. For example, if a business is finding it hard to create invoices for filing GST claims, then students can do a project that takes the raw data (list of transactions), groups the transactions by category, accounts for the GST tax rates, and creates invoices in the appropriate format. Students can be extremely creative here. They can use a wide variety of Python libraries to create user friendly applications such as games, software for their school, software for their disabled fellow students, and mobile applications, of course to do some of these projects, some additional learning is required; this should be encouraged. Students should know how to teach themselves.

The students should be sensitised to avoid plagiarism and violations of copyright issues while working on projects. Teachers should take necessary measures for this.

## Why to use Python:

The following are the primary factors to use python in day-to-day life:

### **1. Python is object-oriented**

Structure supports such concepts as polymorphism, operation overloading and multiple inheritance.

### **2. Indentation**

Indentation is one of the greatest features in python

### **3. It's free (open source)**

Downloading python and installing python is free and easy

### **4. It's Powerful**

- Dynamic typing
- Built-in types and tools
- Library utilities
- Third party utilities (e.g. Numeric, NumPy, SciPy)
- Automatic memory management

### **5. It's Portable**

- Python runs virtually every major platform used today
- As long as you have a compatible python interpreter installed, python programs will run in exactly the same manner, irrespective of platform.

### **6. It's easy to use and learn**

- No intermediate compile
- Python Programs are compiled automatically to an intermediate form called byte code, which the interpreter then reads.
- This gives python the development speed of an interpreter without the performance loss inherent in purely interpreted languages.
- Structure and syntax are pretty intuitive and easy to grasp.

### **7. Interpreted Language**

Python is processed at runtime by python Interpreter

### **8. Interactive Programming Language**

Users can interact with the python interpreter directly for writing the programs

### **9. Straight forward syntax**

The formation of python syntax is simple and straight forward which also makes it popular.

## **Data types:**

The data stored in memory can be of many types. For example, a student roll number is stored as a numeric value and his or her address is stored as alphanumeric characters. Python has various standard data types that are used to define the operations possible on them and the storage method for each of them.

Int  
Float  
Boolean  
String  
List

## **Variables:**

Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory.

Based on the data type of a variable, the interpreter allocates memory and decides what can be stored in the reserved memory. Therefore, by assigning different data types to variables, you can store integers, decimals or characters in these variables.

Rules for Python variables:

- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores(A-z, 0-9, and \_)
- Variable names are case-sensitive (age, Age and AGE are three different variables)

## **Assigning Values to Variables:**

Python variables do not need explicit declaration to reserve memory space. The declaration happens automatically when you assign a value to a variable. The equal sign (=) is used to assign values to variables.

The operand to the left of the = operator is the name of the variable and the operand to the right of the = operator is the value stored in the variable.

### **For example –**

a= 100            # an integer assignment

b = 1000.0        # A floating point

c = "John"        # A string

print (a)

print (b)

print (c)

**This produces the following result –**

100

1000.0

John

### **Multiple Assignment:**

Python allows you to assign a single value to several variables simultaneously. For example :

```
a = b = c = 1
```

Here, an integer object is created with the value 1, and all three variables are assigned to the same memory location. You can also assign multiple objects to multiple variables.

### **For example –**

```
a,b,c = 1,2,"kvsch"
```

Here, two integer objects with values 1 and 2 are assigned to variables a and b respectively, and one string object with the value "john" is assigned to the variable c.

### **Output Variables:**

The Python print statement is often used to output variables.

Variables do not need to be declared with any particular type and can even change type after they have been set.

```
x = 5           # x is of type int
x = "kve "      # x is now of type str
print(x)
```

### **Output: kve**

To combine both text and a variable, Python uses the “+” character:

### **Example**

```
x = "awesome"
print("Python is " + x)
```

### **Output**

Python is awesome

You can also use the + character to add a variable to another variable:

### Example

```
x = "Python is "  
y = "awesome"  
z = x + y  
print(z)
```

### Output:

Python is awesome

### Expressions:

An expression is a combination of values, variables, and operators. An expression is evaluated using assignment operator.

Examples: Y=x + 17

```
>>> x=10  
>>> z=x+20  
>>> z  
30
```

```
>>> x=10  
>>> y=20  
>>> c=x+y  
>>> c  
30
```

A value all by itself is a simple expression, and so is a variable.

```
>>> y=20  
>>> y  
20
```

Python also defines expressions only contain identifiers, literals, and operators. So,

**Identifiers:** Any name that is used to define a class, function, variable module, or object is an identifier.

**Literals:** These are language-independent terms in Python and should exist independently in any programming language. In Python, there are the string literals, byte literals, integer literals, floating point literals, and imaginary literals.

**Operators:** In Python you can implement the following operations using the corresponding tokens.



Operator	Token
add	+
subtract	-
multiply	*
Integer Division	/
remainder	%
Binary left shift	<<
Binary right shift	>>
and	&
or	\
Less than	<
Greater than	>
Less than or equal to	<=
Greater than or equal to	>=
Check equality	==
Check not equal	!=

### Statements:

A statement is an instruction that the Python interpreter can execute. We have normally two basic statements, the assignment statement and the print statement. Some other kinds of statements that are if statements, while statements, and for statements generally called as control flows.

Examples:

An assignment statement creates new variables and gives them values:

```
>>> x=10
>>> school="kve"
```

An print statement is something which is an input from the user, to be printed / displayed on to the screen (or ) monitor.

```
>>> print("kvro ernakulam")
```

```
kv roernakulam
```

## Precedence of Operators:

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.

### Example 1:

```
>>> 3+4*2 11
```

Multiplication gets evaluated before the addition operation

```
>>> (10+10)*2 40
```

Parentheses  $()$  overriding the precedence of the arithmetic operators

### Example 2:

```
a = 20
```

```
b = 10
```

```
c = 15
```

```
d = 5
```

```
e = 0
```

```
e = (a + b) * c / d      #( 30 * 15 ) / 5
print("Value of (a + b) * c / d is ", e)
```

```
e = ((a + b) * c) / d    # (30 * 15) / 5
print("Value of ((a + b) * c) / d is ", e)
```

```
e = (a + b) * (c / d);   # (30) * (15/5)
```



```
print("Value of (a + b) * (c / d) is ", e)
```

```
e = a + (b * c) / d;      # 20 +  
(150/5) print("Value of a + (b * c) / d  
is ", e)
```

### Output:

```
Value of (a + b) * c / d is 90.0  
Value of ((a + b) * c) / d is 90.0  
Value of (a + b) * (c / d) is 90.0  
Value of a + (b * c) / d is 50.0
```

### Comments:

**Single-line comments** begins with a hash(#) symbol and is useful in mentioning that the whole line should be considered as a comment until the end of line.

**A Multi line comment is** useful when we need to comment on many lines. In python, triple double quote(“ “ “) and single quote(' ' ' )are used for multi-line commenting.

### Example:

```
File Edit Format Run Options Window Help  
# Write a python program to add numbers  
a=10 #assigning value to variable a  
b=20 #assigning value to variable b  
""" print the value of a and b using a new variable """  
''' print the value of a and b using a new variable '''  
c=a+b  
print(c)  
|
```

### Output:

```
30
```

### Errors and Exceptions:

**Python Errors and Built-in Exceptions:** Python (interpreter) raises exceptions when it encounters **errors**. When writing a program, we, more often than not, will encounter errors. Error caused by not following the proper structure (syntax) of the language is called syntax error or parsing error

### ZeroDivisionError:

ZeroDivisionError in Python indicates that the second argument used in a division (or modulo) operation was zero.

**OverflowError:**

OverflowError in Python indicates that an arithmetic operation has exceeded the limits of the current Python runtime. This is typically due to excessively large float values, as integer values that are too big will opt to raise memory errors instead.

**ImportError:**

It is raised when you try to import a module which does not exist. This may happen if you made a typing mistake in the module name or the module doesn't exist in its standard path. In the example below, a module named "non\_existing\_module" is being imported but it doesn't exist, hence an import error exception is raised.

**IndexError:**

An IndexError exception is raised when you refer a sequence which is out of range. In the example below, the list abc contains only 3 entries, but the 4th index is being accessed, which will result an IndexError exception.

**TypeError:**

When two unrelated type of objects are combined, TypeErrorexception is raised. In example below, an int and a string is added, which will result in TypeError exception.

**IndentationError:**

Unexpected indent. As mentioned in the "expected an indentedblock" section, Python not only insists on indentation, it insists on consistent indentation. You are free to choose the number of spaces of indentation to use, but you then need to stick with it.

**Syntax errors:**

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

**Run-time error:**

A run-time error happens when Python understands what you are saying, but runs into trouble when following your instructions.

**Key Error :**

Python raises a KeyError whenever a dict() object is requested (using the format a = adict[key]) and the key is not in the dictionary.

## Value Error:

In Python, a value is the information that is stored within a certain object. To encounter a ValueError in Python means that is a problem with the content of the object you tried to assign the value to.

**Python has many built-in exceptions** which forces your program to output an error when something in it goes wrong. In Python, users can define such exceptions by creating a new class. This exception class has to be derived, either directly or indirectly, from Exception class.

## Different types of exceptions:

- ArrayIndexOutOfBoundsException.
- ClassNotFoundException.
- FileNotFoundException.
- IOException.
- InterruptedException.
- NoSuchFieldException.
- NoSuchMethodException

## Modules:

**Modules:** Python module can be defined as a python program file which contains a python code including python functions, class, or variables. In other words, we can say that our python code file saved with the extension (.py) is treated as the module. We may have a runnable code inside the python module. A module in Python provides us the flexibility to organize the code in a logical way. To use the functionality of one module into another, we must have to **import** the specific module.

Syntax:

```
import <module-name>
```

Every module has its own functions, those can be accessed with . (dot)

**Note:** In python we have help ()

Enter the name of any module, keyword, or topic to get help on writing Python programs and using Python modules. To quit this help utility and return to the interpreter, just type "quit".

Some of the modules like os, date, and calendar so on.....

```
>>> import sys
```

```
>>> print (sys.version)
```

```
3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:21:23) [MSC v.1916 32 bit (Intel)]
```

```
>>> print(sys.version_info)
```

```
sys.version_info(major=3, minor=8, micro=0, releaselevel='final', serial=0)
```

```
>>> print(calendar.month(2021,5))
```

```
    May 2021
Mo Tu We Th Fr Sa Su
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
```

```
>>> print(calendar.isleap(2020))
```

```
True
```

```
>>> print(calendar.isleap(2017))
```

```
False
```

## **Control Flow. Loops:**

### **Conditional (if):**

The if statement contains a logical expression using which data is compared and a decision is made based on the result of the comparison.

Syntax:

if expression:  
    statement(s)

If the boolean expression evaluates to TRUE, then the block of statement(s) inside the if statement is executed. If boolean expression evaluates to FALSE, then the first set of code after the end of the if statement(s) is executed.

### **if Statement Flowchart:**

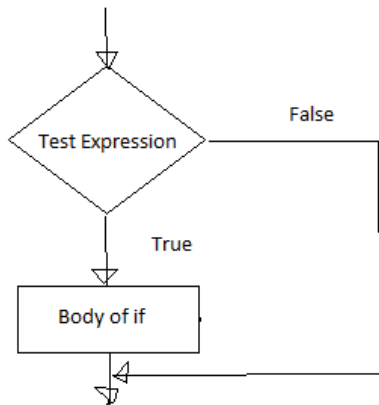


Fig: Operation of if statement

### **Example: Python if Statement**

```
a = 3
if a > 2:
    print(a, "is greater")
print("done")
```

```
a = -1
if a < 0:
    print(a, "a is smaller")
print("Finish")
```

**Output:**

```

3 is greater
done
-1 a is smaller
Finish
----- a=10

```

```

if a>9:

```

```

    print("A is Greater than 9")

```

**Output:**

```

A is Greater than 9

```

**Alternative if (If-Else):**

An else statement can be combined with an if statement. An else statement contains the block of code (false block) that executes if the conditional expression in the if statement resolves to 0 or a FALSE value.

The else statement is an optional statement and there could be at most only one else Statement following if.

**Syntax of if - else :**

```

if test expression:

```

```

    Body of if stmts

```

```

else:

```

```

    Body of else stmts

```

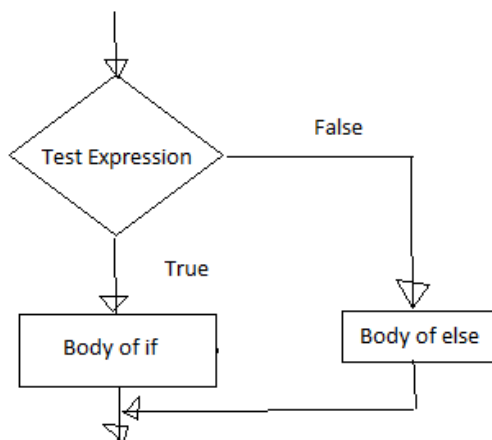
**If - else Flowchart :**

Fig: Operation of if – else statement

**Example of if - else:**

```
a=int(input('enter the number')) if
a>5:
    print("a is greater") else:
    print("a is smaller than the input given")
```

**Output:**

```
enter the number 2
a is smaller than the input given
```

---

```
a=10
b=20
if a>b:
    print("A is Greater than B")
else:
    print("B is Greater than A")
```

**Output:**

```
B is Greater than A
```

**Chained Conditional: (If-elif-else):**

The elif statement allows us to check multiple expressions for TRUE and execute a block of code as soon as one of the conditions evaluates to TRUE. Similar to the else, the elif statement is optional. However, unlike else, for which there can be at most one statement, there can be an arbitrary number of elif statements following an if.

**Syntax of if – elif - else :**

```
If test expression:
    Body of if stmts elif
test expression:
    Body of elif stmts
else:
    Body of else stmts
```



### Flowchart of if – elif - else:

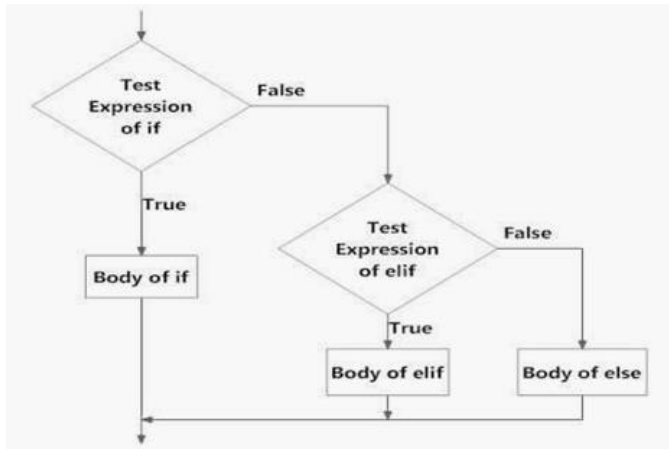


Fig: Operation of if – elif - else statement

### Example of if - elif – else:

```
a=int(input('enter the number'))
b=int(input('enter the number'))
c=int(input('enter the number'))
```

```
if a>b:print("a is greater")
```

```
elif b>c:
    print("b is greater")
```

```
else:
    print("c is greater")
```

### Output:

```
enter the number5
enter the number2
enter the number9
a is greater
```

```
>>>
```

```
enter the number2
enter the number5
enter the number9
c is greater
```

```
-----
var = 100
```

```
if var == 200:
    print("1 - Got a true expression value")
    print(var)
```

```
elif var == 150:
    print("2 - Got a true expression value")
    print(var)
```

```
elif var == 100:
    print("3 - Got a true expression value")
```

```
    print(var)
else:
    print("4 - Got a false expression value")
    print(var)
print("Good bye!")
```

**Output:**

```
3 - Got a true expression value
100
Good bye!
```

## Iteration:

A loop statement allows us to execute a statement or group of statements multiple times as long as the condition is true. Repeated execution of a set of statements with the help of loops is called iteration.

Loops statements are used when we need to run same code again and again, each time with a different value.

## Statements:

In Python Iteration (Loops) statements are of three types:

1. While Loop
2. For Loop
3. Nested For Loops

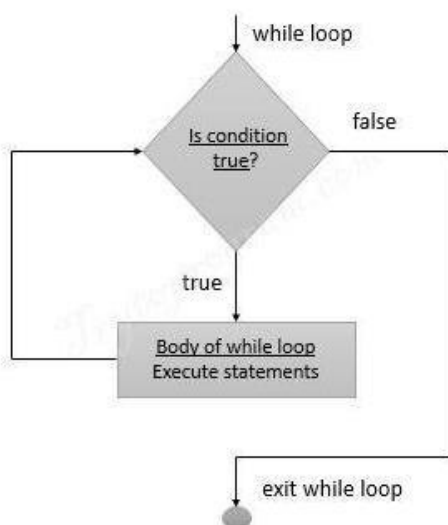
### While loop:

- Loops are either infinite or conditional. Python while loop keeps reiterating a block of code defined inside it until the desired condition is met.
- The while loop contains a boolean expression and the code inside the loop is repeatedly executed as long as the boolean expression is true.
- The statements that are executed inside while can be a single line of code or a block of multiple statements.

### Syntax:

```
while(expression):  
    Statement(s)
```

### Flowchart:



**Example Programs:**

```

1. -----
   i=1
   while i<=6:
       print("KV School") i=i+1

```

**output:**

KV School  
 KV School  
 KV School  
 KV School  
 KV School  
 KV School

```

2. -----
   i=1
   while i<=3:
       print("KV School",end=" ")
       j=1
       while j<=1:
           print("CS DEPT",end="")
           j=j+1
       i=i+1
       print()

```

**Output:**

KV School CS DEPT  
 KV School CS DEPT  
 KV School CS DEPT

## For loop:

Python **for loop** is used for repeated execution of a group of statements for the desired number of times. It iterates over the items of lists, tuples, strings, the dictionaries and other iterable objects

**Syntax:** for var in sequence:

Statement(s)

A sequence of values assigned to var in each iteration

Holds the value of item  
in sequence in each iteration

## Sample Program:

```
numbers = [1, 2, 4, 6, 11, 20]
```

```
seq=0
```

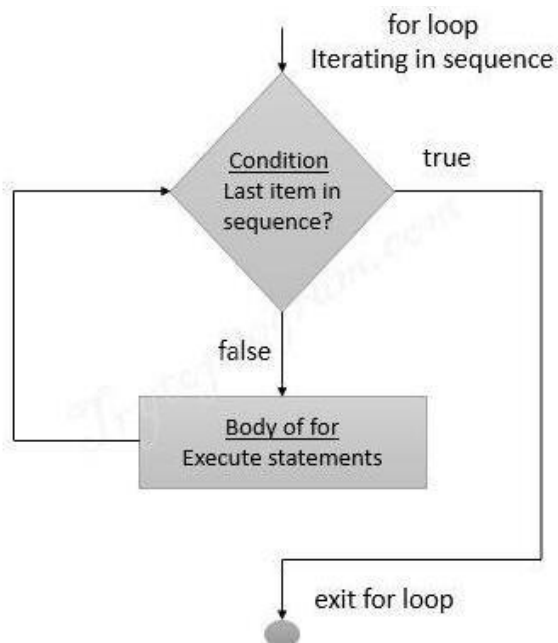
```
for val in numbers:
```

```
    seq=val*val print(seq)
```

## Output:

```
1
4
16
36
121
400
```

## Flowchart:



**Iterating over a list:**

```
#list of items
list = ['K','V','S','C','H']
i = 1
#Iterating over the list
for item in list:
    print ('School ',i,' is ',item)
    i = i+1
```

Output:

```
School 1 is K
School 2 is V
School 3 is S
School 4 is C
School 5 is H
```

**Iterating over a Tuple:**

```
tuple = (2,3,5,7)
print ('These are the first four prime numbers ')
#Iterating over the tuple
for a in tuple:
    print (a)
```

**Output:**

```
These are the first four prime numbers
2
3
5
7
```

**Iterating over a dictionary:**

```
#creating a dictionary
college = {"ces":"block1","it":"block2","ece":"block3"}

#Iterating over the dictionary to print keys
print ('Keys are:')
```

```
for keys in college:  
    print (keys)
```

```
#Iterating over the dictionary to print values  
print ('Values are:')  
for blocks in college.values():  
    print(blocks)
```

Output:  
Keys are:  
ces  
it  
ece

Values are:  
block1  
block2  
block3

### **Iterating over a String:**

```
#declare a string to iterate over  
college = 'KVSCH'
```

```
#Iterating over the string  
for name in college:  
    print (name)
```

### **Output:**

K  
V  
S  
C  
H

### **Nested For loop:**

When one Loop defined within another Loop is called Nested Loops.

### **Syntax:**

```
for val in sequence:  
    for val in sequence:  
        statements  
    statements
```

**# Example 1 of Nested For Loops (Pattern Programs)**

```
for i in range(1,6): for
    j in range(0,i):
        print(i, end=" ")
    print("")
```

**Output:**

```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
-----
```

**# Example 2 of Nested For Loops (Pattern Programs)**

```
for i in range(1,6):
    for j in range(5,i-1,-1):
        print(i, end=" ")
    print("")
```

**Output:**

```
1 1 1 1 1
2 2 2 2
3 3 3
4 4
```



## Break and continue:

In Python, **break** and **continue** statements can alter the flow of a normal loop. Sometimes we wish to terminate the current iteration or even the whole loop without checking test expression. The break and continue statements are used in these cases.

### Break:

The break statement terminates the loop containing it and control of the program flows to the statement immediately after the body of the loop. If break statement is inside a nested loop (loop inside another loop), break will terminate the innermost loop.

### Continue:

The continue statement is used to skip the rest of the code inside a loop for the current iteration only. Loop does not terminate but continues on with the next iteration.

### Pass:

In Python programming, pass is a null statement. The difference between a comment and pass statement in Python is that, while the interpreter ignores a comment entirely, pass is not ignored. Pass is just a placeholder for functionality to be added later.

## Strings:

A string is a group/ a sequence of characters. Since Python has no provision for arrays, we simply use strings. This is how we declare a string. We can use a pair of single or double quotes. Every string object is of the type 'str'.

```
>>> type("name")
<class 'str'>
>>> name=str()
>>>
name ""
>>> a=str('kvsch')
>>> a
' kvsch'
>>> a=str(kvsch)
>>> a[2]
's'
>>> fruit = 'banana'
>>> letter = fruit[1]
```

The second statement selects character number 1 from fruit and assigns it to letter. The expression in brackets is called an index. The index indicates which character in the sequence we want

**String slices:**

A segment of a string is called a slice. Selecting a slice is similar to selecting a character:

Subsets of strings can be taken using the slice operator **([ ] and [:])** with indexes starting at 0 in the beginning of the string and working their way from -1 at the end.

Slice out substrings, sub lists, sub Tuples using index.

**Syntax:[Start: stop: steps]**

- Slicing will start from index and will go up to **stop** in **step** of steps.
- Default value of start is 0,
- Stop is last index of list
- And for step default is 1

**For example 1:**

```
str = 'Hello World!'
print (str) # Prints complete string
print (str[0]) # Prints first character of the string
print (str[2:5]) # Prints characters starting from 3rd to 5th
print (str[2:]) # Prints string starting from 3rd character
print (str * 2) # Prints string two times
print (str + "TEST") # Prints concatenated string
```

**Output:**

```
Hello World! H
llo
llo World!
Hello World!Hello World!
Hello World!TEST
```

**Example 2:**

```
>>> x='computer'
>>> x[1:4]
'omp'

>>> x[1:6:2]
'opt'

>>> x[3:]
```

```
'puter'
>>> x[:5]
'compu'
>>> x[-1]
'r'
>>> x[-3:]
'ter'
>>> x[:-2]
'comput'
>>> x[::-2]
'rtpo'
>>> x[::-1]
'retupmoc'
```

### Immutability:

It is tempting to use the [] operator on the left side of an assignment, with the intention of changing a character in a string.

For example:

```
>>> greeting='kvsch school!'
>>> greeting[0]='n'
```

TypeError: 'str' object does not support item assignment

The reason for the error is that strings are **immutable**, which means we can't change an existing string. The best we can do is creating a new string that is a variation on the original:

```
>>> greeting = 'Hello, world!'
>>> new_greeting = 'J' + greeting[1:]
>>> new_greeting
'Jello, world!'
```

Note: The plus (+) sign is the string concatenation operator and the asterisk (\*) is the repetition operator

## String functions and methods:

There are many methods to operate on String.

S.no	Method name	Description
1.	isalnum()	Returns true if string has at least 1 character and all characters are alphanumeric and false otherwise.
2.	isalpha()	Returns true if string has at least 1 character and all characters are alphabetic and false otherwise.
3.	isdigit()	Returns true if string contains only digits and false otherwise.
4.	islower()	Returns true if string has at least 1 cased character and all cased characters are in lowercase and false otherwise.
5.	isnumeric()	Returns true if a string contains only numeric characters and false otherwise.
6.	isspace()	Returns true if string contains only whitespace characters and false otherwise.
7.	istitle()	Returns true if string is properly "titlecased" and false otherwise.
8.	isupper()	Returns true if string has at least one cased character and all cased characters are in uppercase and false otherwise.
9.	replace(old, new [, max])	Replaces all occurrences of old in string with new or at most max occurrences if max given.
10.	split()	Splits string according to delimiter str (space if not provided) and returns list of substrings;
11.	count()	Occurrence of a string in another string
12.	find()	Finding the index of the first occurrence of a string in another string
13.	swapcase()	Converts lowercase letters in a string to uppercase and viceversa
14.	startswith(str, beg=0, end=len(string))	Determines if string or a substring of string (if starting index beg and ending index end are given) starts with substring str; returns true if so and false otherwise.

## **Output:**

30

## **Array methods:**

Python has a set of built-in methods that you can use on lists/arrays.

Method	Description
<u>append()</u>	Adds an element at the end of the list
<u>clear()</u>	Removes all the elements from the list
<u>copy()</u>	Returns a copy of the list
<u>count()</u>	Returns the number of elements with the specified value
<u>extend()</u>	Add the elements of a list (or any iterable), to the end of the current list
<u>index()</u>	Returns the index of the first element with the specified value
<u>insert()</u>	Adds an element at the specified position
<u>pop()</u>	Removes the element at the specified position
<u>remove()</u>	Removes the first item with the specified value
<u>reverse()</u>	Reverses the order of the list
<u>sort()</u>	Sorts the list

Note: Python does not have built-in support for Arrays, but Python Lists can be used instead.

## LISTS, TUPLES, DICTIONARIES

**Lists:** list operations, list slices, list methods, list loop, mutability, aliasing, cloning lists, list parameters, list comprehension; **Tuples:** tuple assignment, tuple as return value, tuple comprehension; **Dictionaries:** operations and methods, comprehension;

### Lists, Tuples, Dictionaries:

#### List:

- It is a general purpose most widely used in data structures
- List is a collection which is ordered and changeable and allows duplicate members. (Grow and shrink as needed, sequence type, sortable).
- To use a list, you must declare it first. Do this using square brackets and separate values with commas.
- We can construct / create list in many ways.

Ex:

```
>>> list1=[1,2,3,'A','B',7,8,[10,11]]
```

```
>>> print(list1)
```

```
[1, 2, 3, 'A', 'B', 7, 8, [10, 11]]
```

```
>>> x=list()
```

```
>>>
```

```
x []
```

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

```
>>> x=list(tuple1)
```

```
>>> x
```

```
[1, 2, 3, 4]
```

#### List operations:

These operations include indexing, slicing, adding, multiplying, and checking for membership

#### Basic List Operations:

Lists respond to the + and \* operators much like strings; they mean concatenation and repetition here too, except that the result is a new list, not a string.

Python Expression	Results	Description
-------------------	---------	-------------

<code>len([1, 2, 3])</code>	3	Length
<code>[1, 2, 3] + [4, 5, 6]</code>	<code>[1, 2, 3, 4, 5, 6]</code>	Concatenation
<code>['Hi!'] * 4</code>	<code>['Hi!', 'Hi!', 'Hi!', 'Hi!']</code>	Repetition
<code>3 in [1, 2, 3]</code>	True	Membership
<code>for x in [1, 2, 3]: print x,</code>	1 2 3	Iteration

## Indexing, Slicing, and Matrixes

Because lists are sequences, indexing and slicing work the same way for lists as they do for strings.

Assuming following input –

`L = ['kvsch', 'school', 'KVSCH!']`

Python Expression	Results	Description
<code>L[2]</code>	KVSCH	Offsets start at zero
<code>L[-2]</code>	school	Negative: count from the right
<code>L[1:]</code>	<code>['school', 'KVSCH!']</code>	Slicing fetches sections

### List slices:

```
>>> list1=range(1,6)
>>> list1
range(1, 6)
>>> print(list1)
range(1, 6)
>>> list1= [1,2,3,4,5,6,7,8,9,10]
>>> list1[1:]
[2, 3, 4, 5, 6, 7, 8, 9, 10]
>>> list1[:1]
[1]
>>> list1[2:5] [3,
4, 5]
>>> list1[:6] [1,
2, 3, 4, 5, 6]
>>> list1[1:2:4]
[2]
>>> list1[1:8:2]
[2, 4, 6, 8]
```

### List methods:

The list data type has some more methods. Here are all of the methods of list objects:

- Del()
- Append()
- Extend()
- Insert()
- Pop()
- Remove()
- Reverse()
- Sort()

**Delete:** Delete a list or an item from a list

```
>>> x=[5,3,8,6]
>>> del(x[1])          #deletes the index position 1 in a list
>>> x [5,
8, 6]
-----
>>> del(x)
>>> x                  # complete list gets deleted
```

**Append:** Append an item to a list

```
>>> x=[1,5,8,4]
>>> x.append(10)
>>> x
[1, 5, 8, 4, 10]
```

**Extend:** Append a sequence to a list.

```
>>> x=[1,2,3,4]
>>> y=[3,6,9,1]
>>> x.extend(y)
```



```
>>> x
```

```
[1, 2, 3, 4, 3, 6, 9, 1]
```

**Insert:** To add an item at the specified index, use the insert () method:

```
>>> x=[1,2,4,6,7]
```

```
>>> x.insert(2,10)    #insert(index no, item to be inserted)
```

```
>>> x
```

```
[1, 2, 10, 4, 6, 7]
```

```
-----
```

```
>>> x.insert(4,['a',11])
```

```
>>> x
```

```
[1, 2, 10, 4, ['a', 11], 6, 7]
```

**Pop:** The pop() method removes the specified index, (or the last item if index is not specified) or simply pops the last item of list and returns the item.

```
>>> x=[1, 2, 10, 4, 6, 7]
```

```
>>> x.pop() 7
```

```
>>> x
```

```
[1, 2, 10, 4, 6]
```

```
-----
```

```
>>> x=[1, 2, 10, 4, 6]
```

```
>>> x.pop(2)
```

```
10
```

```
>>> x
```

```
[1, 2, 4, 6]
```

**Remove:** The remove() method removes the specified item from a given list.

```
>>> x=[1,33,2,10,4,6]
```

```
>>> x.remove(33)
```

```
>>> x
```

```
[1, 2, 10, 4, 6]
```

```
>>> x.remove(4)
```

```
>>> x
```

```
[1, 2, 10, 6]
```

**Reverse:** Reverse the order of a given list.

```
>>> x=[1,2,3,4,5,6,7]
```

```
>>> x.reverse()
```

```
>>> x
```

```
[7, 6, 5, 4, 3, 2, 1]
```

Sort: Sorts the elements in ascending order

```
>>> x=[7, 6, 5, 4, 3, 2, 1]
```

```
>>> x.sort()
```

```
>>> x
```

```
[1, 2, 3, 4, 5, 6, 7]
```

```
-----
```

```
>>> x=[10,1,5,3,8,7]
```

```
>>> x.sort()
```

```
>>> x
```

```
[1, 3, 5, 7, 8, 10]
```

### List loop:

Loops are control structures used to repeat a given section of code a certain number of times or until a particular condition is met.

#### Method #1: For loop

```
#list of items
```

```
list = ['K','V','S','C','H']
```

```
i = 1
#Iterating over the list
for item in list:
    print ('school ',i,' is ',item)
    i = i+1
```

**Output:**

```
school 1 is K
school 2 is V
school 3 is S
school 4 is C
school 5 is H
```

**Method #2: For loop and range ()**

In case we want to use the traditional for loop which iterates from number x to number y.

```
# Python3 code to iterate over a list
list = [1, 3, 5, 7, 9]
```

```
# getting length of list
length = len(list)
```

```
# Iterating the index
# same as 'for i in range(len(list))'
for i in range(length):
    print(list[i])
```

**Output:**

```
3
5
7
9
```

**Method #3: using while loop**

```
# Python3 code to iterate over a list
list = [1, 3, 5, 7, 9]
```

```
# Getting length of list
length = len(list) i = 0
```

```
# Iterating using while loop while
i < length:
    print(list[i]) i
    += 1
```

### **Mutability:**

A mutable object can be changed after it is created, and an immutable object can't.

**Append:** Append an item to a list

```
>>> x=[1,5,8,4]
>>> x.append(10)
>>> x
[1, 5, 8, 4, 10]
```

**Extend:** Append a sequence to a list.

```
>>> x=[1,2,3,4]
>>> y=[3,6,9,1]
>>> x.extend(y)
>>> x
```

**Delete:** Delete a list or an item from a list

```
>>> x=[5,3,8,6]
>>> del(x[1])          #deletes the index position 1 in a list
>>> x [5,
8, 6]
```

**Insert:** To add an item at the specified index, use the insert () method:

```
>>> x=[1,2,4,6,7]
>>> x.insert(2,10)     #insert(index no, item to be inserted)
```

```
>>> x
```

```
[1, 2, 10, 4, 6, 7]
```

```
-----
```

```
>>> x.insert(4,['a',11])
```

```
>>> x
```

```
[1, 2, 10, 4, ['a', 11], 6, 7]
```

**Pop:** The `pop()` method removes the specified index, (or the last item if index is not specified) or simply pops the last item of list and returns the item.

```
>>> x=[1, 2, 10, 4, 6, 7]
```

```
>>> x.pop() 7
```

```
>>> x
```

```
[1, 2, 10, 4, 6]
```

```
-----
```

```
>>> x=[1, 2, 10, 4, 6]
```

```
>>> x.pop(2)
```

```
10
```

```
>>> x
```

```
[1, 2, 4, 6]
```

**Remove:** The `remove()` method removes the specified item from a given list.

```
>>> x=[1,33,2,10,4,6]
```

```
>>> x.remove(33)
```

```
>>> x
```

```
[1, 2, 10, 4, 6]
```

```
>>> x.remove(4)
```

```
>>> x
```

```
[1, 2, 10, 6]
```

**Reverse:** Reverse the order of a given list.

```
>>> x=[1,2,3,4,5,6,7]
```

```
>>> x.reverse()
```

```
>>> x
```

```
[7, 6, 5, 4, 3, 2, 1]
```

**Sort:** Sorts the elements in ascending order

```
>>> x=[7, 6, 5, 4, 3, 2, 1]
```

```
>>> x.sort()
```

```
>>> x
```

```
[1, 2, 3, 4, 5, 6, 7]
```

```
-----
```

```
>>> x=[10,1,5,3,8,7]
```

```
>>> x.sort()
```

```
>>> x
```

```
[1, 3, 5, 7, 8, 10]
```

### Aliasing:

1. An alias is a second name for a piece of data, often easier (and more useful) than making a copy.
2. If the data is immutable, aliases don't matter because the data can't change.
3. But if data can change, aliases can result in lot of hard – to – find bugs.
4. Aliasing happens whenever one variable's value is assigned to another variable.

**For ex:**

```
a = [81, 82, 83]
```

```

b = [81, 82, 83]
print(a == b)
print(a is b) b
= a
print(a == b)
print(a is b)
b[0] = 5
print(a)

```

**Output:**

```

False
True
True
[5, 82, 83]

```

Because the same list has two different names, a and b, we say that it is **aliased**. Changes made with one alias affect the other. In the example above, you can see that a and b refer to the same list after executing the assignment statement b = a.

### Cloning Lists:

If we want to modify a list and also keep a copy of the original, we need to be able to make a copy of the list itself, not just the reference. This process is sometimes called cloning, to avoid the ambiguity of the word copy.

The easiest way to clone a list is to use the slice operator. Taking any slice of a creates a new list. In this case the slice happens to consist of the whole list.

**Example:**

```

a = [81, 82, 83]
b = a[:]      # make a clone using slice
print(a == b)
print(a is b)
b[0] = 5
print(a)
print(b)

```



**Output:**

True

False

[81, 82, 83]

[5, 82, 83]

Now we are free to make changes to b without worrying about a

**List parameters:**

Passing a list as an argument actually passes a reference to the list, not a copy of the list. Since lists are mutable, changes made to the elements referenced by the parameter change the same list that the argument is referencing.

# for example, the function below takes a list as an argument and multiplies each element in the list by 2:

```
def doubleStuff(List):
```

```
    """ Overwrite each element in aList with double its value. """ for
```

```
    position in range(len(List)):
```

```
        List[position] = 2 * List[position]
```

```
things = [2, 5, 9]
```

```
print(things)
```

```
doubleStuff(things)
```

```
print(things)
```

**Output:**

[2, 5, 9]

[4, 10, 18]

## List comprehension:

### List:

List comprehensions provide a concise way to create lists. Common applications are to make new lists where each element is the result of some operations applied to each member of another sequence or iterable, or to create a subsequence of those elements that satisfy a certain condition.

For example, assume we want to create a list of squares, like:

```
>>> list1=[]  
  
>>> for x in range(10):  
    list1.append(x**2)  
  
>>> list1  
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

(or)

This is also equivalent to

```
>>> list1=list(map(lambda x:x**2, range(10)))  
  
>>> list1  
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

(or)

Which is more concise and readable.

```
>>> list1=[x**2 for x in range(10)]  
  
>>> list1  
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

### **Similarly some examples:**

```
>>> x=[m for m in range(8)]
```

```
>>> print(x)
```

```
[0, 1, 2, 3, 4, 5, 6, 7]
```

```
>>> x=[z**2 for z in range(10) if z>4]
```

```
>>> print(x)
```

```
[25, 36, 49, 64, 81]
```

```
>>> x=[x ** 2 for x in range (1, 11)      if x % 2 == 1]
```

```
>>> print(x)
```

```
[1, 9, 25, 49, 81]
```

```
>>> a=5
```

```
>>> table = [[a, b, a * b] for b in range(1, 11)]
```

```
>>> for i in table:
```

```
    print(i)
```

```
[5, 1, 5]
```

```
[5, 2, 10]
```

```
[5, 3, 15]
```

```
[5, 4, 20]
```

```
[5, 5, 25]
```

```
[5, 6, 30]
```

```
[5, 7, 35]
```

```
[5, 8, 40]
```

```
[5, 9, 45]
```

```
[5, 10, 50]
```

### **Tuples:**

A tuple is a collection which is ordered and unchangeable. In Python tuples are written with round brackets.

- Supports all operations for sequences.
- Immutable, but member objects may be mutable.
- If the contents of a list shouldn't change, use a tuple to prevent items from

accidentally being added, changed, or deleted.

- Tuples are more efficient than list due to python's implementation.

We can construct tuple in many ways:

```
X=()    #no item tuple
```

```
X=(1,2,3)
```

```
X=tuple(list1)
```

```
X=1,2,3,4
```

### Example:

```
>>> x=(1,2,3)
```

```
>>> print(x)
```

```
(1, 2, 3)
```

```
>>> x (1,
```

```
2, 3)
```

```
>>> x=()
```

```
>>> x
```

```
()
```

```
>>> x=[4,5,66,9]
```

```
>>> y=tuple(x)
```

```
>>> y
```

```
(4, 5, 66, 9)
```

```
>>> x=1,2,3,4
```

```
>>> x
```

```
(1, 2, 3, 4)
```

Some of the operations of tuple are:

- Access tuple items
- Change tuple items
- Loop through a tuple
- Count()
- Index()
- Length()

**Access tuple items:** Access tuple items by referring to the index number, inside square brackets

```
>>> x=('a','b','c','g')
```

```
>>> print(x[2]) c
```

**Change tuple items:** Once a tuple is created, you cannot change its values. Tuples are unchangeable.

```
>>> x=(2,5,7,'4',8)
```

```
>>> x[1]=10
```

Traceback (most recent call last):

File "<pyshell#41>", line 1, in <module> x[1]=10

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

```
>>> x
```

```
(2, 5, 7, '4', 8)    # the value is still the same
```

**Loop through a tuple:** We can loop the values of tuple using for loop

```
>>> x=4,5,6,7,2,'aa'
```

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

```
4
```

```
5
```

```
6
```

```
7
```

```
2
```

```
aa
```

**Count ():** Returns the number of times a specified value occurs in a tuple

```
>>> x=(1,2,3,4,5,6,2,10,2,11,12,2)
```

```
>>> x.count(2) 4
```

**Index ():** Searches the tuple for a specified value and returns the position of where it was found

```
>>> x=(1,2,3,4,5,6,2,10,2,11,12,2)
>>> x.index(2) 1
```

(Or)

```
>>> x=(1,2,3,4,5,6,2,10,2,11,12,2)
>>> y=x.index(2)
>>> print(y) 1
```

**Length ():** To know the number of items or values present in a tuple, we use len().

```
>>> x=(1,2,3,4,5,6,2,10,2,11,12,2)
>>> y=len(x)
>>> print(y) 12
```

### Tuple Assignment

Python has tuple assignment feature which enables you to assign more than one variable at a time. In here, we have assigned tuple 1 with the college information like college name, year, etc. and another tuple 2 with the values in it like number (1, 2, 3... 7).

For Example,

Here is the code,

```
>>> tup1 = ('kvsch', 'eng college','2004','cse', 'it','csit');
>>> tup2 = (1,2,3,4,5,6,7);
>>> print(tup1[0])
• kvsch
>>> print(tup2[1:4])
(2, 3, 4)
```

Tuple 1 includes list of information of kvsch

Tuple 2 includes list of numbers in it

We call the value for [0] in tuple and for tuple 2 we call the value between 1 and 4

Run the above code- It gives name kvsch for first tuple while for second tuple it gives number (2, 3, 4)

### **Tuple as return values:**

A Tuple is a comma separated sequence of items. It is created with or without (). Tuples are immutable.

# A Python program to return multiple values from a method using tuple

# This function returns a tuple

```
def fun():
```

```
    str = "kvsch college"
```

```
    x = 20
```

```
    return str, x; # Return tuple, we could also
```

```
                # write (str, x)
```

```
# Driver code to test above method str,
```

```
x = fun() # Assign returned tuple
```

```
print(str)
```

```
print(x)
```

### **Output:**

20

- Functions can return tuples as return values.

```
def circleInfo(r):
```

```
    """ Return (circumference, area) of a circle of radius r """ c = 2
```

```
    * 3.14159 * r
```

```
    a = 3.14159 * r * r
```

```
    return (c, a)
```

```
print(circleInfo(10))
```

### **Output:**

(62.8318, 314.159)

```

def f(x):
    y0 = x + 1
    y1 = x * 3
    y2 = y0 ** y3 return
    (y0, y1, y2)

```

### Tuple comprehension:

Tuple Comprehensions are special: The result of a tuple comprehension is special. You might expect it to produce a tuple, but what it does is produce a special "generator" object that we can iterate over.

#### For example:

```

>>> x = (i for i in 'abc') #tuple comprehension
>>> x
<generator object <genexpr> at 0x033EEC30>

>>> print(x)
<generator object <genexpr> at 0x033EEC30>

```

You might expect this to print as ('a', 'b', 'c') but it prints as <generator object <genexpr> at 0x02AAD710> The result of a tuple comprehension is not a tuple: it is actually a generator. The only thing that you need to know now about a generator now is that you can iterate over it, but ONLY ONCE.

So, given the code

```

>>> x = (i for i in 'abc')
>>> for i in x:
    print(i)

```

```

a
b
c

```

#### Create a list of 2-tuples like (number, square):

```

>>> z=[(x, x**2) for x in range(6)]
>>> z
[(0, 0), (1, 1), (2, 4), (3, 9), (4, 16), (5, 25)]

```



### **Set:**

Similarly to list comprehensions, set comprehensions are also supported:

```
>>> a = {x for x in 'abracadabra' if x not in 'abc'}
```

```
>>> a
```

```
{'r', 'd'}
```

```
>>> x={3*x for x in range(10) if x>5}
```

```
>>> x
```

```
{24, 18, 27, 21}
```

### **Dictionaries:**

A dictionary is a collection which is unordered, changeable and indexed. In Python dictionaries are written with curly brackets, and they have keys and values.

- Key-value pairs
- Unordered

We can construct or create dictionary like: `X={1:'A',2:'B',3:'c'}`

```
X=dict([('a',3) ('b',4)])
```

```
X=dict('A'=1,'B' =2)
```

### **Example:**

```
>>> dict1 = {"brand":"mrcet","model":"college","year":2004}
```

```
>>> dict1
```

```
{'brand': 'mrcet', 'model': 'college', 'year': 2004}
```

### **Operations and methods:**

Methods that are available with dictionary are tabulated below. Some of them have already been used in the above examples.

Method	Description
clear()	Remove all items form the dictionary.

<code>copy()</code>	Return a shallow copy of the dictionary.
<code>fromkeys(seq[, v])</code>	Return a new dictionary with keys from seq and value equal to v (defaults to None).
<code>get(key[,d])</code>	Return the value of key. If key does not exist, return d (defaults to None).
<code>items()</code>	Return a new view of the dictionary's items (key, value).
<code>keys()</code>	Return a new view of the dictionary's keys.
<code>pop(key[,d])</code>	Remove the item with key and return its value or d if key is not found. If d is not provided and key is not found, raises <code>KeyError</code> .
<code>popitem()</code>	Remove and return an arbitrary item (key, value). Raises <code>KeyError</code> if the dictionary is empty.
<code>setdefault(key[,d])</code>	If key is in the dictionary, return its value. If not, insert key with a value of d and return d (defaults to None).
<code>update([other])</code>	Update the dictionary with the key/value pairs from other, overwriting existing keys.
<code>values()</code>	Return a new view of the dictionary's values

**Below are some dictionary operations:**

**To access specific value of a dictionary, we must pass its key,**

```
>>> dict1 = {"brand": "mrcet", "model": "college", "year": 2004}
>>> x = dict1["brand"]
>>> x
'mrcet'
```

**To access keys and values and items of dictionary:**

```
>>> dict1 = {"brand": "mrcet", "model": "college", "year": 2004}
>>> dict1.keys() dict_keys(['brand',
'model', 'year'])
>>> dict1.values() dict_values(['mrcet',
'college', 2004])
>>> dict1.items()
dict_items([('brand', 'mrcet'), ('model', 'college'), ('year', 2004)])
```

```
>>> for items in dict1.values(): print(items)
```

```
mrcet
college
2004
```

```
>>> for items in dict1.keys(): print(items)
```

```
brand
model
year
```

```
>>> for i in dict1.items():
    print(i)
```

```
('brand', 'mrcet')
('model', 'college')
('year', 2004)
```

Some more operations like:

- Add/change

- Remove
- Length
- Delete

**Add/change values:** You can change the value of a specific item by referring to its key name

```
>>> dict1 = {"brand": "mrcet", "model": "college", "year": 2004}
>>> dict1["year"] = 2005
>>> dict1
{'brand': 'mrcet', 'model': 'college', 'year': 2005}
```

**Remove():** It removes or pop the specific item of dictionary.

```
>>> dict1 = {"brand": "mrcet", "model": "college", "year": 2004}
>>> print(dict1.pop("model"))
college
>>> dict1
{'brand': 'mrcet', 'year': 2005}
```

**Delete:** Deletes a particular item.

```
>>> x = {1:1, 2:4, 3:9, 4:16, 5:25}
>>> del x[5]
>>> x
```

**Length:** we use len() method to get the length of dictionary.

```
>>> {1: 1, 2: 4, 3: 9, 4: 16}
{1: 1, 2: 4, 3: 9, 4: 16}
>>> y = len(x)
>>> y
4
```

**Iterating over (key, value) pairs:**

```
>>> x = {1:1, 2:4, 3:9, 4:16, 5:25}
>>> for key in x:
    print(key, x[key])
```

```
1 1
2 4
3 9
```

```
4 16
5 25
>>> for k,v in x.items():
    print(k,v)
```

```
1 1
2 4
3 9
4 16
5 25
```

### List of Dictionaries:

```
>>> customers = [{"uid":1,"name":"John"},
                  {"uid":2,"name":"Smith"},
                  {"uid":3,"name":"Andersson"},
                  ]
>>> >>> print(customers)
[{'uid': 1, 'name': 'John'}, {'uid': 2, 'name': 'Smith'}, {'uid': 3, 'name': 'Andersson'}]
```

## Print the uid and name of each customer

```
>>> for x in customers: print(x["uid"],
                              x["name"])
```

```
1 John
2 Smith
3 Andersson
```

## Modify an entry, This will change the name of customer 2 from Smith to Charlie

```
>>> customers[2]["name"]="charlie"
>>> print(customers)
[{'uid': 1, 'name': 'John'}, {'uid': 2, 'name': 'Smith'}, {'uid': 3, 'name': 'charlie'}]
```

## Add a new field to each entry

```
>>> for x in customers:
    x["password"]="123456" # any initial value
```

```
>>> print(customers)
```

```
[{'uid': 1, 'name': 'John', 'password': '123456'}, {'uid': 2, 'name': 'Smith', 'password': '123456'},
{'uid': 3, 'name': 'charlie', 'password': '123456'}]
```

## Delete a field

```
>>> del customers[1]
>>> print(customers)
[{'uid': 1, 'name': 'John', 'password': '123456'}, {'uid': 3, 'name': 'charlie', 'password':
'123456'}]
```

```
>>> del customers[1]
>>> print(customers)
[{'uid': 1, 'name': 'John', 'password': '123456'}]
```

## Delete all fields

```
>>> for x in customers: del
      x["uid"]
```

```
>>> x
{'name': 'John', 'password': '123456'}
```

### Comprehension:

Dictionary comprehensions can be used to create dictionaries from arbitrary key and value expressions:

```
>>> z={x: x**2 for x in (2,4,6)}
```

```
>>> z
{2: 4, 4: 16, 6: 36}
```

```
>>> dict11 = {x: x*x for x in range(6)}
```

```
>>> dict11
{0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
```

**MCQ  
SECTION A**

1. Which of the following is not a keyword?

- a) eval
- b) assert
- c) nonlocal
- d) pass

**Answer: a**

**Explanation: eval can be used as a variable.**

2. What is the order of precedence in python?

- i) Parentheses
  - ii) Exponential
  - iii) Multiplication
  - iv) Division
  - v) Addition
  - vi) Subtraction
- a) i,ii,iii,iv,v,vi
  - b) ii,i,iii,iv,v,vi
  - c) ii,i,iv,iii,v,vi
  - d) i,ii,iii,iv,vi,v

**Answer: a**

**Explanation: For order of precedence, just remember this PEMDAS (similar to BODMAS).**

3. What error occurs when you execute the following Python code snippet?

```
apple = mango
```

- a) SyntaxError
- b) NameError
- c) ValueError
- d) TypeError

**Answer: b**

**Explanation: Mango is not defined hence name error.**

4. What will be the output of the following Python code snippet?

- 1. def example(a):
- 2. a = a + '2'
- 3. a = a\*2
- 4. return a
- 5. >>>example("hello")

- a) indentation Error
- b) cannot perform mathematical operation on strings

- c) hello2
- d) hello2hello2

**Answer: a**

**Explanation: Python codes have to be indented properly.**

5. What will be the value of X in the following Python expression?

`X = 2+9*((3*12)-8)/10` a)

30.0

b) 30.8

c) 28.4

d) 27.2

**Answer: d**

**Explanation: The expression shown above is evaluated as:  $2+9*(36-8)/10$ , which simplifies to give  $2+9*(2.8)$ , which is equal to  $2+25.2 = 27.2$ . Hence the result of this expression is 27.2.**

6. Select all options that print.

hello-how-are-you

a) `print('hello', 'how', 'are', 'you')`

b) `print('hello', 'how', 'are', 'you' + '-' * 4)`

c) `print('hello-' + 'how-are-you')`

d) `print('hello' + '-' + 'how' + '-' + 'are' + 'you')`

**Answer: c**

**Explanation: Execute in the shell.**

7. Which of the following can be used as valid variable identifier(s) in Python?

a) total

b) 7Salute

c) Que\$tion

d) global

**Answer: a**

8. Which of the following statement is correct for an operator?

a) Python only evaluates the second argument if the first one is False

b) Python only evaluates the second argument if the first one is True

c) Python only evaluates True if any one argument is True

d) Python only evaluates False if any one argument is False

**Answer: b**

9. Which of the following forces an expression to be converted into specific type?

a) Implicit type casting

b) Mutable type casting

c) Immutable type casting

d) Explicit type casting



**Answer: d**

10. Which point can be considered as difference between string and list?

- a) Length
- b) Indexing and Slicing
- c) Mutability
- d) Accessing individual elements

**Answer: c**

11. Which of the following statement is true for extend () list method?

- a) adds element at last
- b) adds multiple elements at last
- c) adds element at specified index
- d) adds elements at random index

**Answer: b**

12. The statement `del l[1:3]` do which of the following task?

- a) deletes elements 2 to 4 elements from the list
- b) deletes 2nd and 3rd element from the list
- c) deletes 1st and 3rd element from the list
- d) deletes 1st, 2nd and 3rd element from the list

**Answer: b**

13. If `l=[11,22,33,44]`, then output of `print(len(l))` will be

- a) 4
- b) 3
- c) 8
- d) 6

**Answer : a**

14. Which of the following method is used to delete element from the list?

- a) `del()`
- b) `delete()`
- c) `pop()`
- d) All of these

**Answer : c**

15 The step argument in `range()` function \_\_\_\_\_.

- a. indicates the beginning of the sequence
- b. indicates the end of the sequence
- c. indicates the difference between every two consecutive numbers in the sequence
- d. generates numbers up to a specified value

**Answer : c**

16. The statement `del L[1:3]` do which of the following task?

- a) Deletes elements 2 to 4 elements from the list
- b) Deletes 2nd and 3rd element from the list
- c) Deletes 1st and 3rd element from the list
- d) Deletes 1st, 2nd and 3rd element from the list

**Answer : b**

### MCQ SECTION B

1. What will be the output of the following Python code?

```
if (9 < 0) and (0 < -9):
    print("hello")
elif (9 > 0) or False:
    print("good")
else:
    print("bad")
```

- a) error
- b) hello
- c) good
- d) bad

**Answer: c**

2. What will be the output of the following Python code?

```
x = ['ab', 'cd']
for i in x:
    i.upper()
print(x)
```

- a) ['ab', 'cd']
- b) ['AB', 'CD']
- c) [None, None]
- d) none of the mentioned

**Answer: a**

**Explanation:** The function `upper()` does not modify a string in place, it returns a new string which isn't being stored anywhere.

3. What will be the output of the following Python code?

```
x = ['ab', 'cd']
for i in x:
```

```
x.append (i.upper ())
```

```
print(x)
```

- a) ['AB', 'CD']
- b) ['ab', 'cd', 'AB', 'CD']
- c) ['ab', 'cd']
- d) none of the mentioned

**Answer: d**

**Explanation:** The loop does not terminate as new elements are being added to the list in each iteration.

4. What will be the output of the following Python code?

```
i = 1
while True:
    if i%3 == 0:
        break
    print(i)
    i += 1
```

- a) 1 2
- b) 1 2 3
- c) error
- d) none of the mentioned

**Answer: c**

**Explanation:** `SyntaxError`, there shouldn't be a space between + and = in +=.

5. What will be the output of the following Python code?

```
string = "my name is x"
for i in string:
    print (i, end=" ",)
```

- a) m, y, , n, a, m, e, , i, s, , x,
- b) m, y, , n, a, m, e, , i, s, , x
- c) my, name, is, x,
- d) error

**Answer: a**

**Explanation:** Variable i takes the value of one character at a time.

6. What will be the output of the following Python code?

```
string = "my name is x"
for i in string.split():
    print (i, end=" ",)
```

- a) m, y, , n, a, m, e, , i, s, , x,
- b) m, y, , n, a, m, e, , i, s, , x
- c) my, name, is, x,

d) error

**Answer: c**

**Explanation: Variable i takes the value of one word at a time.**

7. What will be the output of the following Python code snippet?

```
a = [0, 1, 2, 3]
for a[-1] in a:
    print(a[-1])
```

- a) 0 1 2 3
- b) 0 1 2 2
- c) 3 3 3 3
- d) error

**Answer: b**

**Explanation: The value of a[-1] changes in each iteration.**

8. Which of this a correct statement?

- a) xyz = 10 100 1000
- b) x y z = 10 100 1000
- c) x, y, z = 10, 100, 1000
- d) x y z= 10, 100, 1000

9. Assume the list superstore is [1,2,3,4,5], which of the following is correct syntax for slicing operation?

- a. print(superstore[0:])
- b. print(superstore[:2])
- c. print(superstore[:-2])
- d. All of these

10. If zoo = ["lion", "tiger"], what will be zoo \* 2?

- a. ['lion']
- b. ['lion', 'lion', 'tiger', 'tiger']
- c. ['lion', 'tiger', 'lion', 'tiger']
- d. ['tiger']

11. To add a new element to a list the statement used is?

- a. zoo.add(5)
- b. zoo.append("snake")
- c. zoo.addLast(5)
- d. zoo.addend(4)

12. What will be the output of the following?

```
data = [[[1, 2], [3, 4]], [[5, 6], [7, 8]]]
print(data[1][0][0])
```

- a. 1
- b. 2
- c. 4
- d. 5

13. Consider `phone_book = {"Kalpana":7766554433, "Steffi":4499551100}`. To delete the key "Kalpana" the code used is
- a. `phone_book.delete("Kalpana":7766554433)`
  - b. `phone_book.delete("Kalpana")`
  - c. `del phone_book["Kalpana"]`
  - d. `del phone_book("Kalpana":7766554433)`
14. Assume `d = {"Guido":"Python", "Dennis":"C"}`. To obtain the number of entries in dictionary the statement used is
- a. `d.size()`
  - b. `len(d)`
  - c. `size(d)`
  - d. `d.len()`
15. What will be the output of the following code snippet?
- ```
numbers = {}
letters = {}
comb = {}
numbers[1] = 56
numbers[3] = 7
letters[4] = "B"
comb["Numbers"] = numbers
comb["Letters"] = letters
print(comb)
```
- a. Nested dictionary cannot occur
  - b. `'Numbers': {1: 56, 3: 7}`
  - c. `{'Numbers': {1: 56}, 'Letters': {4: 'B'}}`
  - d. `{'Numbers': {1: 56, 3: 7}, 'Letters': {4: 'B'}}`

**MCQ**  
**SECTION 3: CASE STUDY**

1. Mr. Rahul wants created a dictionary to store the details of his students and to manipulate the data. He wrote a code in Python, help him to complete the code:

```
studentDict = dict() # stmt 1
n = int(input("How Many Students you Want To Input?"))
for i in range(____): # stmt 2 - to enter n number of students details
    rollno = input("Enter Roll No:")
    name = input("Enter Name:")
    physicsMarks = int(input("Enter Physics Marks:"))
    chemistryMarks = int(input("Enter Chemistry Marks:"))
    mathMarks = int(input("Enter Maths Marks:"))
    studentDict[rollno]=[name, physicsMarks, chemistryMarks, mathMarks] # stmt 3
```

- (i) What is the use of stmt 1?

- a) To initialize a dictionary.
- b) To declare an empty dictionary.
- c) To clear the dictionary
- d) None of the above

Answer: b

(ii) Fill in the blank space at stmt 2:

- a. 1,0
- b. n
- c. size(studentDict)
- d. 10

(iii) Write appropriate statement to add data to dictionary, stmt 3.

- a) studentDict[rollno] = [name, physicsMarks, chemistryMarks, mathMarks]
- b) studentDict.add (name, physicsMarks, chemistryMarks, mathMarks)
- c)

2. Consider the following code and answer the questions that follow:

```
Book = { 1: 'Thriller', 2: 'Mystery', 3: 'Crime' , 4: 'Children Stories' }
Library = {'5' : 'Madras Diaries', '6' : 'Malgudi Days' }
```

- (i) Bobby needs to change the title in the dictionary book from 'Crime' to 'Crime Thriller'. He has written the following command:

**Book['Crime'] = 'Crime Thriller'**

But he is not getting the answer. Help him choose the correct command:

- a) Book[2] = 'Crime Thriller'
- b) Book[3] = 'Crime Thriller'
- c) Book[2] = ('Crime Thriller')
- d) Book[3] = ('Crime Thriller')

Answer: b

(ii) The command to merge the dictionary Book with Library the command would be:

- a. d=Book + Library
- b. print (Book + Library)
- c. Book.update(Library)
- d. Library.update(Book)

Answer: c

(iii) What will be the output of the following line of code:

**print ( list ( Library )**

- a) ['5', 'Madras Diaries', '6', 'Malgudi Days']
- b) ('5', 'Madras Diaries', '6', 'Malgudi Days')

c) ['Madras Diaries','Malgudi Days']

d) ['5','6']

Answer: d

- (iv) In order to check whether the key 2 is present in the dictionary Book, Ramesh uses the following command:

### 2 in Book

He gets the answer as 'True'. Now to check whether the name 'Madras Diaries' exist in the dictionary Library, he uses the following command:

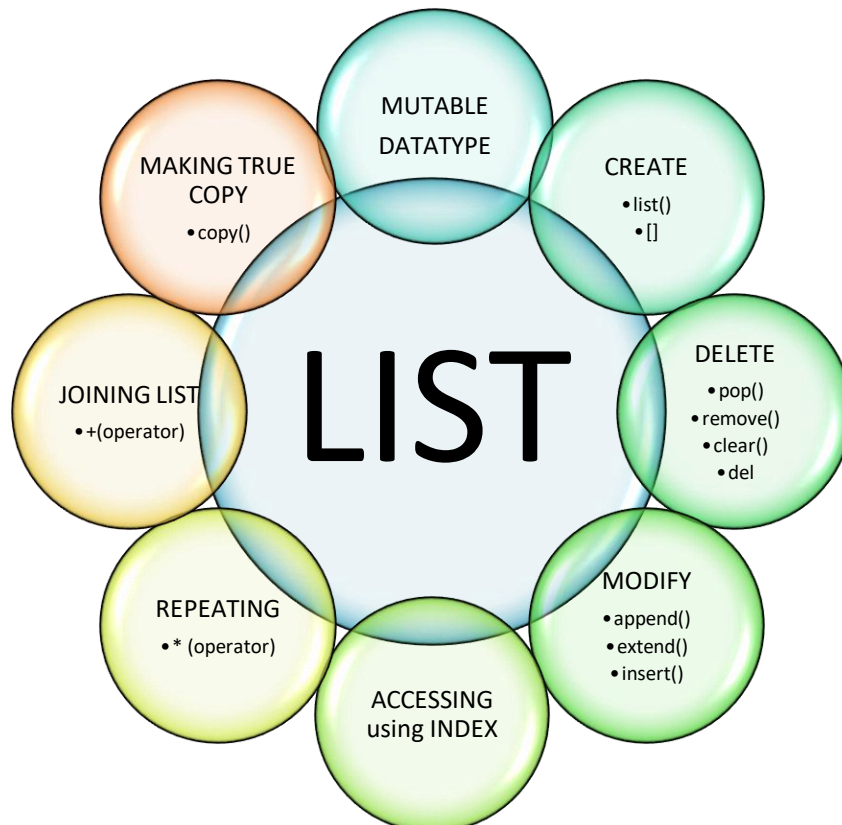
### 'Madras Diaries' in Library

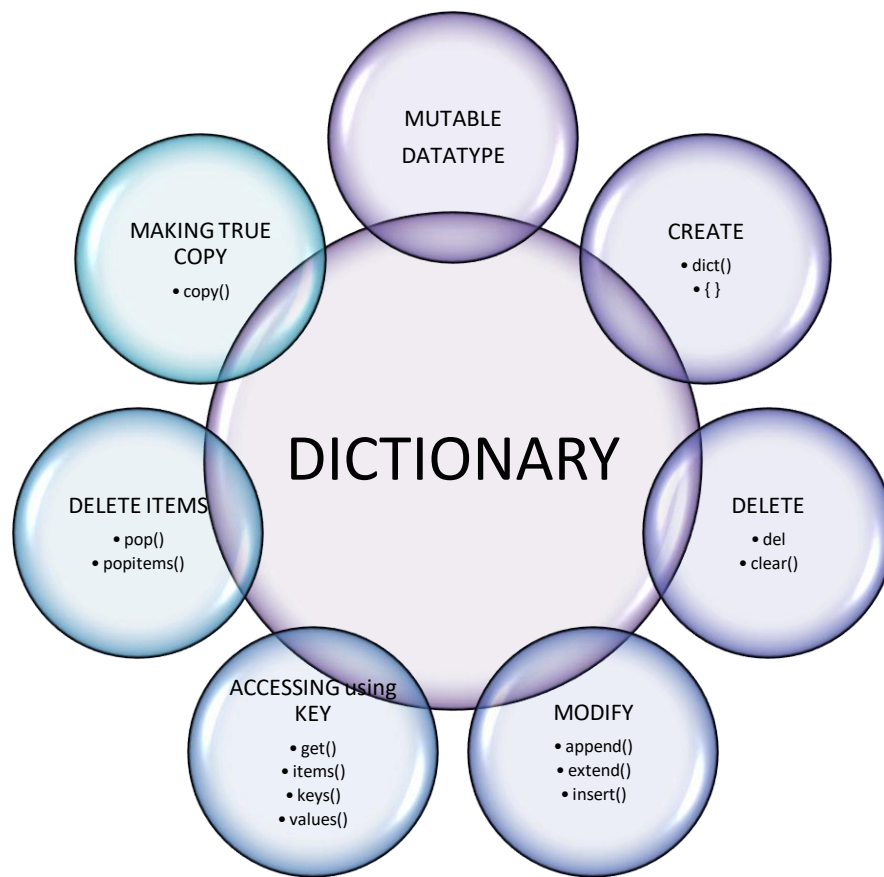
But he gets the answer as 'False'. Select the correct reason for this.

- a. We cannot use the **in** operator with values. It can be used with keys only.
- b. We must use the function Library.values() along with the **in** operator.
- c. We can use the Library.items() function instead of the **in** operator.
- d. Both (b) and (c)

Answer: a

### MIND MAPPING







## FUNCTIONS

A group of statements within a program that perform as specific task.

Usually one task of a large program.

Functions can be executed in order to perform overall program task

Known as divide and conquer approach

### FUNCTION DEFINITION

A function is a named sequence of statement(s) that performs a computation. It contains line of code(s) that are executed sequentially from top to bottom by Python interpreter.

They are the most important building blocks for any software in Python.

**Functions can be categorized as -**

- i. Built-in functions
- ii. Functions defined in module
- iii. User defined functions

#### Built-in functions

🚦 Built in functions are the function(s) that are built into Python and can be accessed by a programmer.

🚦 These are always available and for using them, we don't have to import any module (file).

| Name of the function  | Description                                                                                  | Example                                                 |
|-----------------------|----------------------------------------------------------------------------------------------|---------------------------------------------------------|
| abs (x)               | It returns distance between x and zero, where x is a numeric expression.                     | >>>abs(-45) 45 >>>abs(119L)<br>119                      |
| max( x, y, z, ..... ) | It returns the largest of its arguments: where x, y and z are numeric variable/expression.   | >>>max(80, 100, 1000) 1000<br>>>>max(-80, -20, -10) -10 |
| min( x, y, z, ..... ) | It returns the smallest of its arguments; where x, y, and z are numeric variable/expression. | >>> min(80, 100, 1000) 80 >>><br>min(-80, -20, -10) -80 |

|                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                                                                                            |
|------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>cmp( x, y )</code>                 | It returns the sign of the difference of two numbers: -1 if $x < y$ , 0 if $x == y$ , or 1 if $x > y$ , where $x$ and $y$ are numeric variable/expression.                                                                                                                                                                                                                                                                                                                                                                                                                                                     | <pre>&gt;&gt;&gt;cmp(80, 100) -1 &gt;&gt;&gt;cmp(180, 100) 1</pre>                                                                                                                                                                                                                                                                                         |
| <code>divmod (x,y )</code>               | Returns both quotient and remainder by division through a tuple, when $x$ is divided by $y$ ; where $x$ & $y$ are variable/expression.                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | <pre>&gt;&gt;&gt; divmod (14,5) (2,4) &gt;&gt;&gt; divmod (2.7, 1.5) (1.0, 1.20000)</pre>                                                                                                                                                                                                                                                                  |
| <code>len (s)</code>                     | Return the length (the number of items) of an object. The argument may be a sequence (string, tuple or list) or a mapping (dictionary).                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | <pre>&gt;&gt;&gt; a= [1,2,3] &gt;&gt;&gt;len (a) 3 &gt;&gt;&gt; b= „Hello“ &gt;&gt;&gt; len (b) 5</pre>                                                                                                                                                                                                                                                    |
| <code>range (start, stop[, step])</code> | This is a versatile function to create lists containing arithmetic progressions. It is most often used in for loops. The arguments must be plain integers. If the step argument is omitted, it defaults to 1. If the start argument is omitted, it defaults to 0. The full form returns a list of plain integers $[start, start + step, start + 2 * step, \dots]$ . If step is positive, the last element is the largest $start + i * step$ less than stop; if step is negative, the last element is the smallest $start + i * step$ greater than stop. step must not be zero (or else Value Error is raised). | <pre>&gt;&gt;&gt; range(10)- [0, 1, 2, 3, 4, 5, 6, 7, 8, 9] &gt;&gt;&gt; range(1, 11)- [1,2, 3, 4, 5, 6, 7, 8, 9, 10] &gt;&gt;&gt;range(0, 30, 5)- [0, 5, 10, 15,20, 25] &gt;&gt;&gt; range(0, 10, 3)- [0, 3, 6, 9] &gt;&gt;&gt; range(0, -10, -1)- [0, -1, -2, -3, -4, -5, -6, -7, -8, -9] &gt;&gt;&gt; range(0) - [] &gt;&gt;&gt; range(1, 0) - []</pre> |
| <code>round( x [, n] )</code>            | It returns float $x$ rounded to $n$ digits from the decimal point, where $x$ and $n$ are numeric expressions. If $n$ is not provided then $x$ is rounded to 0 decimal digits.                                                                                                                                                                                                                                                                                                                                                                                                                                  | <pre>&gt;&gt;&gt;round(80.23456, 2) 80.23 &gt;&gt;&gt;round(-100.000056, 3) - 100.0 &gt;&gt;&gt; round (80.23456)</pre>                                                                                                                                                                                                                                    |

|  |  |      |
|--|--|------|
|  |  | 80.0 |
|--|--|------|

## ii. **Functions defined in module**

A module is a file containing Python definitions (i.e., functions) and statements.

Standard library of Python is extended as module(s) to a programmer. Definitions from the module can be used within the code of a program. To use these modules in the program, a programmer needs to import the module.

### **How to import a module**

- ☉ There are many ways to import a module in your program, the one's which you should know are:
- ☉ Import
- ☉ From

### **Import**

It is simplest and most common way to use modules in our code.

Its syntax is:

```
import modulename1 [,modulename2, -----]
```

### **Example**

```
>>> import math
```

To use/ access/invoke a function, you will specify the module name and name of the function-separated by dot (.). This format is also known as dot notation.

Example

```
>>> value= math. sqrt (25) # dot notation
```

### **From Statement**

It is used to get a specific function in the code instead of the complete module file. If we know beforehand which function(s), we will be needing, then we may use from. For modules having large no. of functions, it is recommended to use from instead of import.

Its syntax is:

```
>>> from modulename import functionname [, functionname.....]
```

```
>>>from modulename import * ( Import everything from the file)
```

Example

```
>>> from math import sqrt
```

```
value = sqrt (25)
```

Some functions from math module are :

| Name of the function | Description                                                                         | Example                                                                                                                 |
|----------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| ceil( x )            | It returns the smallest integer not less than x, where x is a numeric expression.   | math.ceil(-45.17)    -45.0<br>math.ceil(100.12)    101.0<br>math.ceil(100.72) 101.0                                     |
| floor( x )           | It returns the largest integer not greater than x, where x is a numeric expression. | math.floor(-45.17)    -46.0<br>math.floor(100.12)    100.0<br>math.floor(100.72) 100.0                                  |
| fabs( x )            | It returns the absolute value of x, where x is a numeric value.                     | math.fabs(-45.17)    45.17<br>math.fabs(100.12)    100.12<br>math.fabs(100.72) 100.72                                   |
| exp( x )             | It returns exponential of x: $e^x$ , where x is a numeric expression.               | math.exp(-45.17)<br>2.41500621326e-20<br>math.exp(100.12)<br>3.03084361407e+43<br>math.exp(100.72)<br>5.52255713025e+43 |

|          |                                                                                   |                                                       |
|----------|-----------------------------------------------------------------------------------|-------------------------------------------------------|
| log( x ) | It returns natural logarithm of x, for $x > 0$ , where x is a numeric expression. | math.log(100.12)<br>4.60636946656<br>math.log(100.72) |
|----------|-----------------------------------------------------------------------------------|-------------------------------------------------------|

|                          |                                                                               |                                                                                                                                                          |
|--------------------------|-------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
|                          |                                                                               | 4.61234438974                                                                                                                                            |
| <code>log10( x )</code>  | It returns base-10 logarithm of x for x > 0, where x is a numeric expression. | <code>math.log10(100.12)</code><br>2.00052084094<br><code>math.log10(100.72)</code><br>2.0031157171                                                      |
| <code>pow( x, y )</code> | It returns the value of xy, where x and y are numeric expressions.            | <code>math.pow(100, 2)</code> 10000.0<br><code>math.pow(100, -2)</code><br>0.0001 <code>math.pow(2, 4)</code> 16.0<br><code>math.pow(3, 0)</code> 1.0    |
| <code>sqrt( x )</code>   | It returns the square root of x for x > 0, where x is a numeric expression.   | <code>math.sqrt(100)</code> -10.0<br><code>math.sqrt(7)</code> -<br>2.6457513110645907                                                                   |
| <code>cos( x )</code>    | It returns the cosine of x in radians, where x is a numeric expression        | <code>math.cos(3)</code> -0.9899924966<br><code>math.cos(-3)</code> -0.9899924966<br><code>math.cos(0)</code> 1.0<br><code>math.cos(math.pi)</code> -1.0 |
| <code>sin( x )</code>    | It returns the sine of x, in radians, where x must be a numeric value.        | <code>math.sin(3)</code> 0.14112000806<br><code>math.sin(-3)</code> -<br>0.14112000806<br><code>math.sin(0)</code> -0.0                                  |
| <code>tan( x )</code>    | It returns the tangent of x in radians, where x must be a numeric value.      | <code>math.tan(3)</code> -<br>0.142546543074<br><code>math.tan(-3)</code><br>0.142546543074<br><code>math.tan(0)</code> 0.0                              |

|             |                                                                               |                                                                                                                                         |
|-------------|-------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| degrees (x) | It converts angle x from radians to degrees, where x must be a numeric value. | <code>math.degrees(3)</code><br>171.887338539<br><code>math.degrees(-3)</code><br>171.887338539<br><code>math.degrees(0)</code> 0.0     |
| radians(x)  | It converts angle x from degrees to radians, where x must be a numeric value. | <code>math.radians(3)</code><br>0.0523598775598<br><code>math.radians(-3)</code><br>0.0523598775598<br><code>math.radians(0)</code> 0.0 |

Some functions from random module are :

| Name of the function                           | Description                                                    | Example                                                                                                                    |
|------------------------------------------------|----------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| <code>random ( )</code>                        | It returns a random float x, such that $0 \leq x$              | <code>&gt;&gt;&gt;random.random( )</code><br>0.281954791393<br><code>&gt;&gt;&gt;random.random( )</code><br>0.309090465205 |
| <code>randint (a, b)</code>                    | It returns a int x between a & b such that $a \leq x \leq b$   | <code>&gt;&gt;&gt; random.randint (1,10)</code> 5<br><code>&gt;&gt;&gt; random.randint (2,20)</code><br>-1                 |
| <code>uniform (a,b)</code>                     | It returns a floating point number x, such that $a \leq x < b$ | <code>&gt;&gt;&gt;random.uniform (5, 10)</code><br>5.52615217015                                                           |
| <code>randrange ([start,] stop [,step])</code> | It returns a random item from the given range                  | <code>&gt;&gt;&gt;random.randrange(100,1000,3)</code> 150                                                                  |

Some functions from statistics module are :

| Name of the function | Description | Example |
|----------------------|-------------|---------|
|----------------------|-------------|---------|

|           |                                           |                                                                                           |
|-----------|-------------------------------------------|-------------------------------------------------------------------------------------------|
| mean(x)   | It returns arithmetic mean                | >>>statistics.mean([11,24,32,45,51]) 32.6                                                 |
| median(x) | It returns median (middle value) of x     | >>>statistics.median([11,24,32,45,51]) 32                                                 |
| mode(x)   | It returns mode (the most repeated value) | >>>statistics.mode([11,24,11,45,11]) -11<br>>>>statistics.mode(("red","blue","red"))'red' |

Some of the other modules, which you can explore, are: string, time, date.

### iii. User defined functions

- ❖ To define a function keyword def is used
- ❖ After the keyword comes an identifier i.e. name of the function, followed by parenthesized list of parameters and the colon which ends up the line.
- ❖ Next follows the block of statement(s) that are the part of function.

### Creating User Defined Function

#### Syntax to define a user defined function:

def <function\_name> ( [ parameters] ): # **def is a keyword**

statement1 # **parameters are optional**

statement2 # **function name is an identifier**

-----

return <value> # **here return is a keyword**

Example:

```
def area_of_circle( radius ):
```

```
    print("Area of circle = ", 3.1416*radius*radius)
```

```
r=int(input("Enter radius of circle"))
```

```
area_of_circle(r)
```

**output:**

Enter radius of circle 10

Area of circle = 314.16

### Points about User Defined Functions:

- Every User Defined Function start with the Keyword def.
- The first line of the function is called Function Definition.
- Function name must follow norms of valid identifier.
- Every function header end with parenthesis and colon sign followed by function name
- We can provide parameters inside parenthesis, which is optional.
- The return statement is used to return a value from function. The return statement is also optional. The function should return a value only at the place from where it get's called.
- To use a function, it must be called or invoked in main program.

### Type of user defined functions:

The user defined functions can be categorized in following 4 categories:

- 1.Function which accept arguments and return some value
- 2.Function which accept arguments but return no value
- 3.Functions which do not accept any argument but return some value
- 4.Functions which do not accept any argument and even not return any value

**Parameters/Arguments passed to functions:** These are the values passed to user defined functions from the calling function. If you are passing values of immutable types (i.e., numbers, strings,etc) to the Called function, then the called function cannot alter their values.

- If you are passing values of mutable types (i.e., List, Dictionary, etc.) to the Called function, then the called function can make changes to them.

### They are categorized in following two types:

- Formal Parameters or Formal Arguments: These are the values provided in Parenthesis when we write a function Header.
- Actual Parameters or Actual Arguments: These are the values passed to the function when it is called/invoked.



Example:

```
def area_of_circle( radius ): # here radius is the Formal Parameter or Formal Argument
    print("Area of circle = ",3.1416*radius*radius)
r=int(input("Enter radius of circle"))
ar=area_of_circle(r) # here r is the Actual Parameter or Actual Argument
```

output:

Enter radius of circle 10

Area of circle = 314.16

### Types of Arguments:

In Python following four types of arguments allowed:

- ❖ Positional Arguments
- ❖ Default Arguments
- ❖ Keyword Arguments
- ❖ Variable length Arguments

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)**

### (i) Default Parameters

The parameters which are assigned with a value in function header while defining the function are known as default parameters. This value is optional for the parameter. If a user explicitly passes the value in function call, then the value which is passed by the user will be taken by the default parameter. If no value is provided, then the default value will be taken by the parameter. Default parameters will be written in the end of the function header, means **positional parameter cannot appear to the right side of default parameter.**

Example:

```
def SIcal(amount, rate, time=10):      #Function definition header
```

## (ii) Required Arguments /Positional Parameters

When the function call statement must match the number and positional order of arguments as defined in the function definition, this is called the positional parameters.

Example:

```
def Test(x,y,z): #function definition header
```

Then we can call function using following statements

```
p,q,r=3,4,6
```

```
Test(p,q,r)      #3 variables are passed
```

```
Test(4,q,r)      # 1 literal and 2 variables are passed
```

```
Test(5,6,7)      # 3 literals are passed.
```

So x,y,z are positional parameters and the value must be provided to these parameters

## (iii) Keyword arguments:

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. This allows you to place argument out of order because the Python interpreter is able to use the keywords provided to match the values with parameters.

```
def show(x,y):
    print("Name is ",x)
    print("Age is ",y)
show(y=18,x="Raj")
```

.

OUTPUT

```
Name is Raj
```

```
Age is 18
```

**(iv) Variable Number of Arguments:**

In cases where you don't know the exact number of arguments that you want to pass to a function, you can use the following syntax with `*args`: The variable stored in `*args` are represented in the form of tuple.

```
def change(*a):
    for i in a:
        print(i)
    print(a)
change(10,5)
change(15,20,25)
```

**OUTPUT**

```
10
5
(10, 5)
15
20
25
(15, 20, 25)
```

**(v) Arguments are passed as a dictionary (\*\*kwargs):**

Kwargs allow you to pass keyword arguments to a function. They are used when you are not sure of the number of keyword arguments that will be passed in the function. Kwargs can be used for unpacking dictionary key, value pairs. This is done using the double asterisk notation ( `**` ).

In the function, we use the double asterisk `**` before the parameter name to denote this type of argument. The arguments are passed as a dictionary and these arguments make a dictionary inside function with name same as the parameter excluding double asterisk `**`

Example:

```
def fun(**data):
```

```

print("\nData type of argument:",type(data))
print(data)
for key, value in data.items():
    print(key,value)
fun(Name="Aman",Age=18,Marks=56)

```

#### OUTPUT

```

Data type of argument: <class 'dict'>
{'Name': 'Aman', 'Age': 18, 'Marks': 56}
Name Aman
Age 18
Marks 5

```

- **A function can return multiple values. The return values should be a comma-separated list of values and Python then constructs a tuple and returnsthis to the caller.**
- **Having a positional argument after keyword areguments will result into error**
- **What is the difference between parameter and argument?**

Ans. Parameters are temporary variable names within functions. The argument can be thought of as the value that is assigned to that temporary variable. For instance, lets consider the following simple function to calculate sum of two numbers.

```

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

```

a,b here are the parameter for the function 'sum'.

Arguments are used in procedure calls, i.e. the values passed to the function at run-time.10,20 are the arguments for the function sum

#### SCOPE OF VARIABLES:

Scope of a variable is the portion of a program where the variable is recognized and can be accessed therein.

There are two types of scope for variables:

1. Local Scope
2. Global Scope

### 1. Local Scope:

A variable declared in a function-body is said to have local scope. It can not be accessed outside the function. In this scope, the lifetime of variables inside a function is as long as the function executes.

### 2. Global Scope:

A variable declared in top level segment (`__main__`) of a program is said to have a global scope.

In this scope, lifetime of a variable inside a program is as long as the program executes.

Example 1:

```
def Sum(x , y) :
    z = x + y
    return z

a = 5
b = 7
s = Sum(a , b)
print(s)
```

In the above program x, y and z are local variables and are destroyed once we return from the function. a, b and s are global variables and remain in the memory until the program executes.

Example 2:

```
def my_func( ):
    x=10          # local variable
    print("Value inside function:",x)
x=20             # global variable
my_func( )
```

```
print("Value outside function:",x)
```

Output:

Value inside function:10

Value outside function:20

Here, we can see that the value of x is 20 initially. Even though the function my\_func() changed the value of x to 10, it did not affect the value outside the function. This is because the variable x inside the function is different (local to the function) from the one outside. Although they have same names, they are two different variables with different scope. On the other hand, variables outside of the function are visible from inside. They have a global scope.

## RANDOM FUNCTION

- **randint()**

randint(x,y) will return a value  $\geq x$  and  $\leq y$ . It will generate a random number from the **inclusive range**.

**eg: random.randint(0,50) will return a random number between 0 and 50 (both inclusive)**

- **randrange()**

randrange(x,y) will return a value  $\geq x$  and  $< y$ . The randrange(start, stop, step) doesnot include the stop number while generating random integer. ie. **stop is exclusive**.

**eg. random.randrange(0,50) will return a random number between 0 and 49.**

## SECTION A

|   |                                                                                                                                    |
|---|------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Which of the following is a valid function name?<br><b>a)Start_game()</b> b) start game()   c) start-game()    d) All of the above |
| 2 | . If the return statement is not used in the function then which type of value will be returned by thefunction?                    |

|    | a) int                      b) str                      c) float                      d) <b>None</b>                                                                                                                                                                                                          |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3  | Richa is working with a program where she gave some values to the function. She doesn't know the term to relate these values. Help her by selecting the correct option.<br>a) function value                      b) <b>arguments or parameters</b><br>c) return values                      d) function call |
| 4  | What is the minimum and maximum value of c in the following code snippet?<br><br>import random<br>a = random.randint(3,5)<br>b = random.randint(2,3)<br>c = a + b<br>print(c)<br><br>a) 3 , 5                      b) <b>5, 8</b> c) 2, 3                      d) 3, 3                                        |
| 5  | In python function, the function calling another function is known as _____ and the function being called is known _____<br><br>a) main, keyword                      b) <b>caller, called</b><br>c) called, caller                      d) executer, execute                                                 |
| 6  | What will be the output of the following code ?<br><br>print("100+200")<br><br>a) 300                      b) <b>100200</b> c) 100+200                      d) 200                                                                                                                                            |
| 7  | pow( ) function belongs to which library ?<br><br>a) <b>math</b> b) string                      c) random                      d) maths                                                                                                                                                                       |
| 8  | What data type is the object below?<br><br>L = (1, 23, 'hello',1)<br><br>a) list                      b) dictionary                      c) array                      d) <b>tuple</b>                                                                                                                        |
| 9  | What is returned by int(math.pow(3, 2))?<br>a) 6    b) <b>9</b> c) error, third argument required    d) error, too many arguments                                                                                                                                                                             |
| 10 | Which of the following is not a type conversion functions?<br><br>a) Int()    b) Str()                      c) <b>Input()</b> d) Float()                                                                                                                                                                      |
| 11 | Write the output of the following:<br><br>print(float())                                                                                                                                                                                                                                                      |

|    |                                                                                                                                                                                                                                                                     |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | a) 1.0 <b>b) 0.0</b> c) Any floating-point number      d) zero                                                                                                                                                                                                      |
| 12 | Identify the module to which the following function load ( ) belong to?<br>a) math    b) random <b>c) pickle</b> d) sys                                                                                                                                             |
| 13 | How many argument(s) a function can receive<br>a) Only one    b) 0 or many    c) Only more than one    d) At least one                                                                                                                                              |
| 14 | Give the output<br><pre>def fun():     global a     a=10     print(a)  a=5 fun() print(a)</pre><br>a) 10                      b) 5                      c) 5                      d) 10<br>10                      10                      5                      5 |
| 15 | Value returning functions should be generally called from inside of an expression<br>a) True      b) False                                                                                                                                                          |
| 16 | The variable declared inside the function is called a _____ variable<br>a) global      b) local      c) external      d) none of the above                                                                                                                          |
| 17 | These are predefined functions that are always available for use. For using them we don't need to import any module<br>a) built in function                      b) pre define function<br>c) user defined function                      d) none of the above       |
| 18 | The _____ of a variable is the area of the program where it may be referenced<br>a) external      b) global      c) scope      d) local                                                                                                                             |
| 19 | If you want to communicate between functions ie. calling and called statement, then you should use _____<br>a) values    b) return    c) arguments                      d) none of the above                                                                        |
| 20 | Which of the following function header is correct<br>a) def mul(a=2, b=5,c)                      b) def mul(a=2, b, c=5)                                                                                                                                            |



|     | c) def mul(a, b=2, c=5)      b) def mul(a=2, b, c=5)                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 21  | <p>Find the flow of execution of the following code :</p> <pre> 1.   def calculate (a, b): 2.       res=a**b 3.       return res 4. 5.   def study(a): 6.       ans=calculate(a,b) 7.       return ans 8. 9.   n=2 10.  a=study(n) 11.  print(a) </pre> <p>a) 1 &gt; 5 &gt; 9 &gt; 10 &gt; 6 &gt; 2 &gt; 3 &gt; 7 &gt; 11    b) 5 &gt; 9 &gt; 10 &gt; 6 &gt; 2 &gt; 3 &gt; 7 &gt; 11<br/> c) 9 &gt; 10 &gt; 5 &gt; 1 &gt; 6 &gt; 2 &gt; 3 &gt; 7 &gt; 11      d) None of the above</p> |
| 22. | <p>Python resolves the scope of a name using the LEGB rule</p> <p>a) True      b) False</p>                                                                                                                                                                                                                                                                                                                                                                                            |
| 23  | <p>A void function internally returns legal empty value _____</p> <p>a) None      b) Close()      c) Return      d) all</p>                                                                                                                                                                                                                                                                                                                                                            |
| 24  | <p>When you use multiple type argument in function, then default argument take place _____</p> <p>a) at beginning    b) at end    c) anywhere    d) none of the above</p>                                                                                                                                                                                                                                                                                                              |
| 25  | <p>A _____ can be skipped in the function call statements</p> <p>a) named parameter      b) default parameter<br/> c) keyword parameters    d) all of the above</p>                                                                                                                                                                                                                                                                                                                    |
| Ans | <p>13. b) 0 or many</p> <p>14. a)      10<br/>                 10</p> <p>15. a) True</p>                                                                                                                                                                                                                                                                                                                                                                                               |

|                                            |
|--------------------------------------------|
| 16. b) local                               |
| 17. a) built in function                   |
| 18. c) scope                               |
| 19. c) arguments                           |
| 20. c) def mul(a, b=2, c=5)                |
| 21. a) 1 > 5 > 9 > 10 > 6 > 2 > 3 > 7 > 11 |
| 22. a) True                                |
| 23. a) None                                |
| 24. b) at end                              |
| 25. b) default parameter                   |

## SECTION B

1. What will be the output of the following Python code?

```
def cube(x):
```

```
    return x * x * x
```

```
x = cube(3)
```

```
print x
```

- a) 9                      b) 3                      **c) 27**                      d) 30

2. Which of the following items are present in the function header?

- a) function name  
b) parameter list  
c) return value  
**d) Both A and B**

3. What is the output of the following snippet?

```
def fun (n):
```

```
    if (n > 100):
```

```
        return n - 5
```

```
    return fun (fun
```

```
(n+11))
```

print (fun (45))

- a) 50      **b) 100**      c)74      d) Infinite loop

4. Predict the output of the following code

```
def func1(list1):
    for x in list1:
        print(x.lower(),end="#")
func1(['New','Dehli'])
```

- A. [New,Dehli]  
**B. new#dehli#**  
 C. newdehli#  
 D. New#Dehli#

5. What will be the output of the following python code:

```
def mul (num1, num2):
    x = num1 * num2
x = mul(20,30)
```

- A. 600    B. None      **C. No Output**      D. 0

6. Which of the following function header is Correct :

- A. def fun(x=1,y)  
 B. def fun(x=1,y,z=2)  
**C. def fun(x=1,y=1,z=2)**  
 D. def fun(x=1,y=1,z=2,w)

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

```
x = 50
def func (x) :
    x = 2
```

```
func (x)
print ('x is now', x)
```

- A) x is now 50      B) x is now 2      C) x is now 100      D) Error

8. Choose the correct output from the options given below.

```
print('Welcome!')
print('Iam',__name__) # __ is double underscore
```

|                                            |                                            |
|--------------------------------------------|--------------------------------------------|
| <p>a)</p> <pre>Welcome! Iam __main__</pre> | <p>b)</p> <pre>Welcome! Iam __name__</pre> |
| <p>c)</p> <pre>Welcome! Iam __abc__</pre>  | <p>d) None of the above</p>                |

Answer a)

9. Find the output

```
def update(x=10):
    x+=15
    print("x=",x)
x=20
update()
print("x=",x)
```

|                            |                            |
|----------------------------|----------------------------|
| <p>a)</p> <pre>25 20</pre> | <p>b)</p> <pre>25 25</pre> |
|----------------------------|----------------------------|

|    |          |                      |
|----|----------|----------------------|
| c) | 20<br>25 | d) None of the above |
|----|----------|----------------------|

Answer c)

10. Predict the output of the following code fragment.

```
def display(x=2,y=3):
    x=x+y
    y += 2
    print(x,y)
display()
display(5,1)
display(9)
```

|    |                    |    |                    |
|----|--------------------|----|--------------------|
| a) | 5 5<br>6 3<br>12 5 | b) | 12 5<br>5 6<br>6 3 |
| c) | 5 6<br>12 5<br>6 3 | d) | None of the above  |

Answer b)

11. Find the output

```
print(pow(5,4,9))
```

a)7      b)0      **c)4**      d)error

|    |                                                                                                                                                   |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------|
| 12 | <p>Give the output of the following program</p> <pre>def check(a):     for i in range(len(a)):         a[i]=a[i]+5     return a b=[1,2,3,4]</pre> |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------|

|    |                                                                                                                                                                                                                                                                                                                                                             |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <p>c=check(b)<br/>print(c)</p> <p>a) [6, 8, 8, 9]                      b) [6, 7, 8, 9]<br/>c) [7, 7, 8, 9]                      d) [6, 7, 9, 9]</p> <p>Ans. b) [6, 7, 8, 9]</p>                                                                                                                                                                             |
| 13 | <p>Give the output</p> <pre>def abc(x,y=60):     return x+y a=20 b=30 a=abc(a,b) print(a,b) b=abc(a) print(a,b) a=abc(b) print(a,b)</pre> <p>a) 50 30 50 110 170                      b) 30 50 110 170 110<br/>c) 50 30                                      d) 50 30 50 110 170 110<br/>50 110<br/>170 110</p> <p>Ans. c) 50 30<br/>50 110<br/>170 110</p> |
| 14 | <pre>def Execute(M):     if M%3==0:         return M*3     else:</pre>                                                                                                                                                                                                                                                                                      |

|    |                                                                                                                                                                                                                                                                                                                                                                                                                                |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <pre> return M+10;  def Output(B=2):     for T in range (0,B):         print(Execute(T),"*",end="")     print()  Output(4) Output() Output(3)  Ans. 0 *11 *12 *9 * 0 *11 * 0 *11 *12 * </pre>                                                                                                                                                                                                                                  |
| 15 | <p>Find the output of the following program :</p> <pre> def Changelt(Text,C):     T=""     for K in range(len(Text)):         if Text[K]&gt;='F' and Text[K]&lt;='L':             T=T+Text[K].lower();         elif Text[K]=='E' or Text[K]=='e':             T=T+C;         elif K%2==0:             T=T+Text[K].upper()         else:             T=T+T[K-1]     print(T)  OldText="pOwERALone" Changelt(OldText,"%") </pre> |

|    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <p>(a) PPW%RLLN%                      (b) PPWCRRINC</p> <p>(c) PPW%RIIN%                      (d) PpW%RrIIN%</p> <p>Ans: c) PPW%RIIN%</p>                                                                                                                                                                                                                                                                                                                                                                      |
| 16 | <p>Find the output of the following code</p> <pre>def disp(str):     m=' '     for i in range(0,len(str)):         if(str[i].isupper()):             m=m+str[i].lower()         elif str[i].islower():             m=m+str[i].upper()         else:             if i%2==0:                 m=m+str[i-1]             else:                 m=m+"@"     print(m.swapcase()) disp('StudyBag\$2021')</pre> <p>a) StudyBagG@2@2   b) sTUDYbAGg@2@2   c) StudyBagG\$2\$2   d) None</p> <p>Ans : a) StudyBagG@2@2</p> |
| 17 | <p>What will be the output of the following code</p> <pre>total=0 def add(a,b):     global total     total=a+b     print(total) add(6,6) print(total)</pre>                                                                                                                                                                                                                                                                                                                                                    |



|    |                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <p>a) 12      b) 12      c) 0      d) None</p> <p>12          0          12</p> <p>Ans : a)</p>                                                                                                                                                                                                                                                                                                                                             |
| 18 | <p>Find the output of the following codes</p> <pre>import random print(int(15+random.random()*5),end=' ') print(int(15+random.random()*5),end=' ') print(int(15+random.random()*5),end=' ') print(int(15+random.random()*5))</pre> <p>a) 19 15 16 19      b) 19 20 16 19      c) 20 14 16 19      d) 19 19 19 19</p> <p>Ans : (a) and (d)</p>                                                                                               |
| 19 | <p>Choose correct answer</p> <pre>def fun1(num):     return num+5 print(fun1(5)) print(num)</pre> <p>a) Print value 10      b) Print value 5      c) Name Error      d) 25</p> <p>Ans c)</p>                                                                                                                                                                                                                                                |
| 20 | <p>Choose the correct option:</p> <p>Statement1: Local Variables are accessible only within a function or block in which it is declared.</p> <p>Statement2: Global variables are accessible in the whole program.</p> <p>a. Statement1 is correct but Statement2 is incorrect</p> <p>b. Statement2 is correct but Statement1 is incorrect</p> <p>c. Both Statements are Correct</p> <p>d. Both Statements are incorrect</p> <p>Ans : c)</p> |
| 21 | <p>Consider the following code and choose correct answer:</p>                                                                                                                                                                                                                                                                                                                                                                               |

|    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <pre>def nameage(name, age):     return [age,name] t=nameage('kishan',20) print(type(t))</pre> <p>a) tuple      b) list      c) (kishan,20)      d) None of all</p> <p>Ans : b)</p>                                                                                                                                                                                                                                                                                      |
| 22 | <p>Consider the code given below and Identify how many times the message “<b>Hello All</b>” will be printed</p> <pre>def prog(name):     for x in name:         if x.isalpha():             print('Alphabet')         elif x.isdigit():             print('Digit')         elif x.isupper():             print('Capital Letter')         else:             print('Hello All') prog('vishal123@gmail.com')</pre> <p>a) 0      b) 2      c) 1      d) 3</p> <p>Ans: b)</p> |
| 23 | <p>Write the output of the following:</p> <pre>a=(10, 12, 13, 12, 13, 14, 15) print(max(a) + min(a) + a.count(2))</pre> <p>a) 13      b) 25      c) 26      d) Error</p> <p>Ans : b)</p>                                                                                                                                                                                                                                                                                 |
| 24 | <p>What will be the output for the below code snippet:</p> <pre>def div(lst,n):     for i in range(0,n):         if lst[i]%5==0:</pre>                                                                                                                                                                                                                                                                                                                                   |

|    |                                                                                                                                                                                                                                                                                                                                      |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <pre>lst[i]+=5 else:     lst[i]=lst[i]/2 lt=[45,20,23,54,5] div(lt, len(lt)) for i in lt:     print(i,end='#')</pre> <p>a) 50#25#11.5#27.0#10#                      b) 50#25#11#27#10#<br/> c) 50#25#1#0#10#                              d) 225#100#1#0#25#<br/> Ans : b)</p>                                                       |
| 25 | <p>What will be the correct output for the following code after removing the syntactical error. Mention the identified error.</p> <pre>def check(x=4, y):     if x != y:         return x+5     else:         return y+10 print(check(10,5))</pre> <p>a) 15              b) 20              c) 5              d) 10<br/> Ans. a)</p> |

### SECTION C

fig 1

```
def myfun4(mylist):
    new=[3,5]
    mylist=new
    mylist.append(6)
    print("list within called function after all changes",mylist)    #LINE 1
    return
list1=[1,4]
print("list before function call",list1)    #LINE 2
myfun4(list1)
print("list after function call",list1)    #LINE 3
```

Based on the code (fig 1) given above answer following questions.

1. Give the output of LINE 1

- a) **list within called function after all changes [3,5,6]**
- b) list within called function after all changes[1,4,6]
- c) list before function call [1,4,3,5,6]
- d) error

2. Give the output of LINE 2

- a. list before function call [1,4,6]
- b. list before function call [1,4]**
- c. list before function call [3,5,6]
- d. Value Error

3. Give the output of LINE 3

- a) list after function call [3,5,6]
- b) list after function call [1,4]**
- c) list after function call [1,4,6]
- d) Value Error

# **DATA FILE HANDLING**

## DATA FILE HANDLING

**File-** A file is a sequence of bytes on the disk/permanent storage where a group of related data is stored. File handling in Python enables us to create, update, read, and delete the files stored on the file system through our python program.

Data File handling takes place in the following order.

- 1- Opening a file.
- 2- Performing operations (read, write) or processing data.
- 3- Closing the file.

### **Types of File in Python:**

Python allows us to create and manage three types of data files.

- 1- Text file
- 2- Binary file
- 3- CSV file

**Text file:** A text file is simply a sequence of ASCII or Unicode characters. A line is a sequence of characters, stored on permanent storage. In a text file, each line is terminated by a special character, known as End Of Line (EOL). Text file can be created using any text editor. Ex. Myfile.txt.

**Binary file:** A binary file stores the data in the same way as stored in the memory. The .exe files, mp3 file, image files, word documents are some of the examples of binary files. We can't read a binary file using a text editor.

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

### **Compare text files, binary files and csv files**

| Sl.no | Text file                                                                       | Binary file                                                        | CSV file                                                        |
|-------|---------------------------------------------------------------------------------|--------------------------------------------------------------------|-----------------------------------------------------------------|
| 1     | It is capable to handle textual data.                                           | It is capable to handle large file.                                | It is very common format and platform independent.              |
| 2     | It consists of series of lines of a set of letters, numbers or symbols (String) | It consists of data with a specific pattern without any delimiter. | It consists of plain text with a list of data with a delimiter. |

|   |                                                         |                                                                                        |                                                                               |
|---|---------------------------------------------------------|----------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|
| 3 | Any text editors like notepad can be used to read them. | No specific programs can be used to read them, python provides functions to read data. | It can be read using text editors like notepads and spreadsheet software.     |
| 4 | Every line ends with EOL.                               | There is no specific EOL character.                                                    | It terminates a line automatically when the delimiter is not used after data. |

**Absolute Path** – It is a full path of the file from the root directory. Absolute paths ensure that Python can find the exact file on your computer.

Ex : - C:\Users\Tanmay\Desktop\Delete\file\_handling.txt

**Relative Path** – It is the path of the file from the current working directory.

Relative Path is the hierarchical path that locates a file or folder on a file system starting from the current directory. The relative path is different from the absolute path, which locates the file or folder starting from the root of the file system.

Ex.

/home/users/c/computerhope/public\_html/cgi-bin- **Absolute path**

./public\_html/cgi-bin – **Relative path**

## Operation On Files In Python

1.Opening a file– open() function

2.Read/Write file–

– Read Functions – read() , readline(), readlines()

– Write Functions – write() , writelines()

3.Close the File – close() Function

4.Remove the file – remove() Function of OS Module

## Read The Contents Of An Existing File

### 1.Open the file–

```
file_object = open("read_sample.txt", 'r') # Relative Path
```

```
file_object = open("C:\Desktop\Notes\sample.txt") #Absolute Path
```

### 2.Reading the file–

```
rf = file_object.read()
```

```
print(rf)
```

### 3.Closing the file–

```
file_object.close()
```

### Working of open() function

We use open ( ) function in Python to open a file in read or write mode. open ( ) will return a file object. To return a file object we use open() function along with two arguments, that accepts file name and the mode, whether to read or write. So, the syntax being: open(filename, mode). There are three kinds of mode, that Python provides and how files can be opened:

“ r “, for reading.

“ w “, for writing.

“ a “, for appending.

### Modes Description

1. r - Opens a file for reading only. The file pointer is placed at the beginning of the file. This is the default mode. Gives error if file does not exist.
2. r+ - Opens a file for both reading and writing. The file pointer placed at the beginning of the file.
3. W - 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. w+ - 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. A - 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. a+ - 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.

### Opening a file using with clause

In Python, we can also open a file using with clause.



The syntax of with clause is:

```
with open (file_name, access_mode) as file_object:
```

The advantage of using with clause is that any file that is opened using this clause is closed automatically, once the control comes outside the with clause. In case the user forgets to close the file explicitly or if an exception occurs, the file is closed automatically. Also, it provides a simpler syntax.

```
with open("myfile.txt","r+") as myObject:
```

```
content = myObject.read()
```

Here, we don't have to close the file explicitly using close() statement. Python will automatically close the file.

### **Reading from a file**

There are three ways to read data from a text file.

**read()** : Returns the read bytes in form of a string. Reads n bytes, if no n specified, reads the entire file.

```
File_object.read([n])
```

**readline()** : Reads a line of the file and returns in form of a string. For specified n, reads at most n bytes. However, does not reads more than one line, even if n exceeds the length of the line.

```
File_object.readline([n])
```

**readlines()** : Reads all the lines and return them as each line a string element in a list.

```
File_object.readlines()
```

Note: '\n' is treated as a special character of two bytes.

### **Writing onto text files:**

#### **write() function**

The write() function will write the content in the file without adding any extra characters.

```
file_name.write(content)
```

#### **writelines() function**

This function writes the content of a list to a file.

```
file_name.writelines(list_of_lines)
```

```
# Program to show various ways to read data from a file.

# Creating a file
file1 = open("myfile.txt", "w")
L = ["This is Delhi \n", "This is Paris \n", "This is London \n"]

# Writing data to a file
file1.write("Hello \n")
file1.writelines(L)
file1.close() # to change file access modes

file1 = open("myfile.txt", "r+")

print("Output of Read function is ")
print(file1.read())
print()

# seek(n) takes the file handle to the nth
# byte from the beginning.
file1.seek(0)

print("Output of Readline function is ")
print(file1.readline())
print()

file1.seek(0)

# To show difference between read and readline
print("Output of Read(9) function is ")
print(file1.read(9))
print()

file1.seek(0)

print("Output of Readline(9) function is ")
print(file1.readline(9))
print()

file1.seek(0)
```

```
# readlines function
print("Output of Readlines function is ")
print(file1.readlines())
print()
file1.close()
```

**Output:**

Output of Read function is

Hello

This is Delhi

This is Paris

This is London

Output of Readline function is

Hello

Output of Read(9) function is

Hello

Th

Output of Readline(9) function is

Hello

Output of Readlines function is

['Hello \n', 'This is Delhi \n', 'This is Paris \n', 'This is London \n']

**Setting Offsets in a File**

if we want to access data in a random fashion, then Python gives us seek() and tell() functions to do so

### The tell() method

This function returns an integer that specifies the current position of the file object in the file. The position so specified is the byte position from the beginning of the file till the current position of the file object. The syntax of using tell() is:

```
file_object.tell()
```

### The seek() method

This method is used to position the file object at a particular position in a file. The syntax of seek() is:

```
file_object.seek(offset [, reference_point])
```

In the above syntax, offset is the number of bytes by which the file object is to be moved. reference\_point indicates the starting position of the file object. That is, with reference to which position, the offset has to be counted. It can have any of the following values:

0 - beginning of the file

1 - current position of the file

2 - end of file

By default, the value of reference\_point is 0, i.e. the offset is counted from the beginning of the file.

For example, the statement `fileObject.seek(5,0)` will position the file object at 5th byte position from the beginning of the file.

### MCQ ON DATA FILE HANDLING

|    |                                                                                                                                                                                                                                      |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. | <p>Which statement will read 5 characters from a file(file object 'f')?</p> <p>a. <code>f.read()</code><br/> b. <code>f.read(5)</code><br/> c. <code>f.reads(5)</code><br/> d. None of the above</p> <p><b>Correct Answer: b</b></p> |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|    |                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2  | <p>Default EOL character in Python</p> <ul style="list-style-type: none"> <li>a. <code>\n</code></li> <li>b. <code>\r</code></li> <li>c. <code>"</code></li> <li>d. <code>\t</code></li> </ul> <p><b>Correct Answer: a</b></p>                                                                                                                                                                                                      |
| 3  | <p>Which of the following is not a file extension for text files?</p> <ul style="list-style-type: none"> <li>a. .txt</li> <li>b. .ini</li> <li>c. .rtf</li> <li>d. .DAT</li> </ul> <p><b>Correct Answer: d</b></p>                                                                                                                                                                                                                  |
| 4  | <p>What is a file object?</p> <ul style="list-style-type: none"> <li>a. It serves as a link to the file.</li> <li>b. It is a file present in a computer.</li> <li>c. A keyword</li> <li>d. A module in python</li> </ul> <p><b>Correct Answer: a</b></p>                                                                                                                                                                            |
| 5  | <p>Which of the following command is used to open a file "c:\temp.txt" in read-mode only?</p> <ul style="list-style-type: none"> <li>a. <code>infile = open("c:\temp.txt", "r")</code></li> <li>b. <code>infile = open("c:\\temp.txt", "r")</code></li> <li>c. <code>infile = open(file = "c:\temp.txt", "r+")</code></li> <li>d. <code>infile = open(file = "c:\\temp.txt", "r+")</code></li> </ul> <p><b>Correct answer:b</b></p> |
| 6  | <p>Which of the following command is used to open a file "c:\temp.txt" in write-mode only?</p> <ul style="list-style-type: none"> <li>a. <code>outfile = open("c:\\temp.txt", "rw")</code></li> <li>b. <code>outfile = open("c:\temp.txt", "w+")</code></li> <li>c. <code>outfile = open("c:\\temp.txt", "w")</code></li> <li>d. <code>outfile = open("c:\\temp.txt", "a")</code></li> </ul> <p><b>Correct answer:c</b></p>         |
| 7. | <p>What does the <code>&lt;readlines()&gt;</code> method returns?</p> <ul style="list-style-type: none"> <li>a.str</li> <li>b.a list of lines</li> </ul>                                                                                                                                                                                                                                                                            |

|     |                                                                                                                                                                            |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|     | c.list of single characters<br>d.list of integers<br><br><b>Correct answer:b</b>                                                                                           |
| 8.  | What is the last action that must be performed on a file?<br>a.save<br>b.close<br>c.end<br>d.write<br><br><b>Correct answer:b</b>                                          |
| 9.  | A file object is also known as<br>a.File handle<br>b.File copy<br>c.File directory<br>d.File link<br><br><b>Correct answer:a</b>                                           |
| 10. | How to open a text file in write and read mode?<br>a. r+<br>b. a+<br>c. wr<br>d. wb<br><br><b>Correct answer:a</b>                                                         |
| 11. | Syntax for closing a file:<br>a. closefile(<file object>)<br>b. <fileobject>.close()<br>c. <filename>.closer()<br>d. closefile.<fileobject><br><br><b>Correct answer:b</b> |
| 12. | Which method can not be used to read from files?<br>a. read()<br>b. readlines()<br>c. readlines(<filename>)<br>d. readline()                                               |

|     |                                                                                                                                                                                                                                                                                                  |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|     | <b>Correct answer:c</b>                                                                                                                                                                                                                                                                          |
| 13. | <p>What does strip() function do?</p> <ul style="list-style-type: none"> <li>a. Removes the trailing or leading spaces, if any.</li> <li>b. Deletes the file</li> <li>c. Remove the file object</li> <li>d. Removes all the spaces between words</li> </ul> <p><b>Correct answer:a</b></p>       |
| 14. | <p>When reading a file using the file object, what method is best for reading the entire file into a single string?</p> <ul style="list-style-type: none"> <li>a. readline()</li> <li>b. read_file_to_str()</li> <li>c. read()</li> <li>d. readlines()</li> </ul> <p><b>Correct answer:c</b></p> |
| 15. | <p>Which file can open in any text editor and is in human readable form?</p> <ul style="list-style-type: none"> <li>a. Binary files</li> <li>b. Text files</li> <li>c. Data files</li> <li>d. Video files</li> </ul> <p><b>Correct answer:b</b></p>                                              |
| 16. | <p>Which function breaks the link of file-object and the file on the disk?</p> <ul style="list-style-type: none"> <li>a.close()</li> <li>b.open()</li> <li>c.tell()</li> <li>d.readline()</li> </ul> <p><b>Correct answer:a</b></p>                                                              |
| 17. | <p>Which function reads the leading and trailing spaces along with trailing newline character ('\n') also while reading the line?</p> <ul style="list-style-type: none"> <li>a.readlines()</li> <li>b.readline()</li> <li>c.read()</li> <li>d.flush()</li> </ul> <p><b>Correct answer:b</b></p>  |
| 18. | <p>Which mode is used to retain its previous data and allowing to add new data?</p> <ul style="list-style-type: none"> <li>a. write mode</li> <li>b. read mode</li> <li>c. open mode</li> <li>d. append mode</li> </ul>                                                                          |

|     |                                                                                                                                                                                                    |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|     | <b>Correct answer:d</b>                                                                                                                                                                            |
| 19. | Which function forces the writing of data on disc still pending in output buffer?<br>a. seek()<br>b. tell()<br>c. flush()<br>d. write()<br><b>Correct answer:c</b>                                 |
| 20. | Syntax for flush() function is:<br>a.<fileObject>(flush())<br>b.flush().<fileobject><br>c.<fileObject>.flush()<br>d.flush().<file-object><br><b>Correct answer:c</b>                               |
| 21  | Which function returns the entire file content in a list where each line is one item of the list?<br>a. readlines()<br>b. readline()<br>c. output()<br>d. Input()<br><b>Correct answer:a</b>       |
| 22  | Which function is used to remove the given character from trailing end i.e. right end?<br>a. strip()<br>b. remove()<br>c. lstrip()<br>d.rstrip()<br><b>Correct answer:d</b>                        |
| 23  | Sometimes the last lap of data remains in buffer and is not pushed onto disk until a _____ operation is performed.<br>a. dump()<br>b. close()<br>c. load()<br>d. open()<br><b>Correct answer:b</b> |
| 24. | The position of a file-pointer is governed by the_____.<br>a. File mode<br>b. append mode<br>c. write mode<br>d. open mode<br><b>Correct answer:a</b>                                              |



|     |                                                                                                                                                                                                                                                                                                                                                |
|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 25. | <p>In which mode the file must exist already, otherwise python raises an error?</p> <ul style="list-style-type: none"> <li>a. read mode</li> <li>b. write mode</li> <li>c. binary mode</li> <li>d. None of these</li> </ul> <p><b>Correct answer:a</b></p>                                                                                     |
| 26. | <p>What is the prefix r stands for in file path?</p> <ul style="list-style-type: none"> <li>a. raw string</li> <li>b. read</li> <li>c. write</li> <li>d. append</li> </ul> <p><b>Correct answer:a</b></p>                                                                                                                                      |
| 27. | <p>In which mode_____if the file does not exist, then the file is created?</p> <ul style="list-style-type: none"> <li>a. read write mode</li> <li>b. read mode</li> <li>c. write mode</li> <li>d. All of these</li> </ul> <p><b>Correct answer:c</b></p>                                                                                       |
| 28. | <p>Which option is correct about this program?</p> <pre>f=open("ss.txt","wb") print("Name of the file:",f.name) f.flush() f.close()</pre> <ul style="list-style-type: none"> <li>a. Compilation error</li> <li>b. Runtime error</li> <li>c. No output</li> <li>d. Flushes the file when closing them</li> </ul> <p><b>Correct answer:d</b></p> |
| 29. | <p>What is the output of the following?</p> <pre>import sys sys.stdout.write('Hello\n') sys.stdout.write('Python\n')</pre> <ul style="list-style-type: none"> <li>a. error</li> <li>b. Runtime error</li> <li>c. Hello Python</li> <li>d. Hello<br/>Python</li> </ul> <p><b>Correct answer:d</b></p>                                           |

|     |                                                                                                                                                                                                                                                      |
|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 30  | <p>Which function is used to read all the characters in text files?</p> <ul style="list-style-type: none"> <li>a. read()</li> <li>b. readcharacters()</li> <li>c. readall()</li> <li>d. readchar()</li> </ul> <p><b>Correct answer:a</b></p>         |
| 31  | <p>Which function is used to read all the lines?</p> <ul style="list-style-type: none"> <li>a. read()</li> <li>b. readall()</li> <li>c. readlines()</li> <li>d. readline()</li> </ul> <p><b>Correct answer:c</b></p>                                 |
| 32  | <p>In which format does the readlines() function give the output?</p> <ul style="list-style-type: none"> <li>a. Integer type</li> <li>b. list type</li> <li>c. string type</li> <li>d. tuple type</li> </ul> <p><b>Correct answer:b</b></p>          |
| 33  | <p>In which format does the read() function give the output?</p> <ul style="list-style-type: none"> <li>a. Integer type</li> <li>b. string type</li> <li>c. list type</li> <li>d. tuple type</li> </ul> <p><b>Correct answer:b</b></p>               |
| 34  | <p>Which function is used to write a list of strings in a file?</p> <ul style="list-style-type: none"> <li>a. writestatement()</li> <li>b. writelines()</li> <li>c. writefulline()</li> <li>d. writeline()</li> </ul> <p><b>Correct answer:b</b></p> |
| 35  | <p>Which function is used to write all the characters?</p> <ul style="list-style-type: none"> <li>a. writechar()</li> <li>b. writecharacters()</li> <li>c. write()</li> <li>d. writeall()</li> </ul> <p><b>Correct answer:c</b></p>                  |
| 36. | <p>What is the correct syntax of open() function?</p> <ul style="list-style-type: none"> <li>a. file=open(file_name[,access_mode][,buffering])</li> </ul>                                                                                            |

|     |                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|     | b. fileobject=open(file_name[,access_model][,buffering])<br>c. fileobject=filename.open()<br>d. none of the mentioned<br><b>Correct answer:b</b>                                                                                                                                                                                                                                                          |
| 37  | In file handling, what does means "r", "a"?<br>a. append, read<br>b. read, append<br>c. read, add<br>d. None of the mentioned<br><b>Correct answer:b</b>                                                                                                                                                                                                                                                  |
| 38  | The default file open mode is....<br>a. w<br>b. r+<br>c. w+<br>d. r<br><b>Correct answer:d</b>                                                                                                                                                                                                                                                                                                            |
| 39. | What is the difference between r+ and w+ modes?<br>a. In r+ mode, file length truncates to zero.<br>b. In w+ mode, file length truncates to zero either file exists or not.<br>c. No difference<br>d. Depends on the operating system<br><b>Correct answer:b</b>                                                                                                                                          |
| 40. | A file maintains a _____ which tells the current position in the file where writing or reading will take place.<br>a. line<br>b. file pointer<br>c. list<br>d. order<br><b>Correct answer:b</b>                                                                                                                                                                                                           |
| 41  | Which of the following statements is true regarding the opening modes of a file?<br>a. While opening a file for reading, if the file does not exist, an error occurs.<br>b. While opening a file for writing ,if the file does not exist, an error occurs.<br>c. While opening a file for reading, if the file does not exist, a new file is created.<br>d. None of the above.<br><b>Correct answer:a</b> |
| 42  | To force python to write the contents of file buffer on to storage file, ..... method may be used.<br>a. buffer()<br>b. flush()<br>c. close()<br>d. write()<br><b>Correct answer:b</b>                                                                                                                                                                                                                    |

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 43  | <p>A binary file "Book.dat" has structure [BookNo, Book_Name, Author, Price].<br/> i. Write a user defined function CreateFile() to input data for 3 records and add to Book.dat .<br/> Your teachers intentionally kept few blanks in between the code and asked you to fill the blanks so that the code will run to find desired result. Do the needful with the following python code.</p> <pre>def CreateFile(): import ..... #line1 def CreateFile():     file =open("d:\\Book.dat",'.....') #line2     bno=input('enter book number')     bname=input('enter book name')     author=input('enter author name')     price=input('enter price')     record=[bno,bname,author,price]     ..... #line3     print('record inserted')     file.close() CreateFile()</pre> <p>Correct option to fill the line 1</p> <p>a.csv<br/> b.CSV<br/> c.PICKLE<br/> d.pickle</p> <p><b>Correct answer:d</b></p> |
| 44. | <p>Choose correct statement for line 2 .</p> <p>a.wb<br/> b.ab<br/> c.w<br/> d.a</p> <p><b>Correct answer:b</b></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 45  | <p>Select correct statement to write data into file for line3 .</p> <p>a.pickle.dump(record,file)<br/> b.pickle.dump(record)<br/> c.pickle.dump(file,record)<br/> d.pickle.load(record,file)</p> <p><b>Correct answer:a</b></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |  |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| 46. | <p>Your teacher has given you a method/function FilterWords() in python which read lines from a text file NewsLetter.TXT, and display those words, which are lesser than 4 characters. Your teachers intentionally kept few blanks in between the code and asked you to fill the blanks so that the code will run to find desired result. Do the needful with the following python code.</p> <pre> def FilterWords():     c=0     file=open('NewsLetter.TXT', '____') #Statement-1     line = file.____ #Statement-2     word = ____ #Statement-3     for c in word:         if ____: #Statement-4             print(c)     ____ #Statement-5  FilterWords() </pre> |  |
|     | <p>Fill in the blank in statement-1</p> <p>a)w<br/>b).ab<br/>c)a<br/>d).r</p> <p><b>Correct answer:d</b></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |  |
| 47. | <p>Fill in the blank in statement-2 to read the data from the file.</p> <p>a.File.Read()<br/>b.file.read()<br/>c.read.lines()<br/>d.readlines()</p> <p><b>Correct answer:b</b></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |  |
| 48. | <p>Fill in the blank in statement-3 to read data word by word.</p> <p>a.Line.Split()<br/>b.Line.split()<br/>c.line.split()<br/>d.split.word()</p> <p><b>Correct answer:c</b></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |  |
| 49. | <p>Fill in the blank in statement-4, which display the word having lesser than 4characters</p> <p>a.len(c) ==4<br/>b.len(c)&lt;4<br/>c.len() = =3<br/>d.len() =3</p> <p><b>Correct answer:d</b></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |  |

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 50. | <p>Your teacher has given you a method/function AMCount() in Python, which should read each character of a text file STORY.TXT, should count and display the occurrence of alphabets A and M (including small cases a and m too). Your teachers intentionally kept few blanks in between the code and asked you to fill the blanks so that the code will run to find desired result. Do the needful with the following python code.</p> <pre>def AMCount():     file=open("story.txt","r")     data=file..... #line1     ca=0     cm=0     for i in .....: #line2         if i in.....: #line3             ca=ca+1             if i in ['M','m']:                 cm=cm+1     file.close()     print('A or a =',ca)     print('M or m =',cm)     ..... #line4 to call that function</pre> <p>choose correct option for line 1</p> <p>a.readline()<br/>b.read(file)<br/>c.read()<br/>d.readlines()</p> <p><b>Correct answer:c</b></p> |
| 51. | <p>choose correct option for line 2</p> <p>a.file<br/>b.data<br/>c.story<br/>d.data.split()</p> <p><b>Correct answer:b</b></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 52. | <p>choose correct option for line 3</p> <p>a.Aa<br/>b.['A','a']<br/>c.aA<br/>d.data</p> <p><b>Correct answer:b</b></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 53. | <p>choose correct option for line 4</p> <p>a.file.AMCount()<br/>b.AMCount<br/>c.AMCount()<br/>d.AMCount(story.txt)</p> <p><b>Correct answer:c</b></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

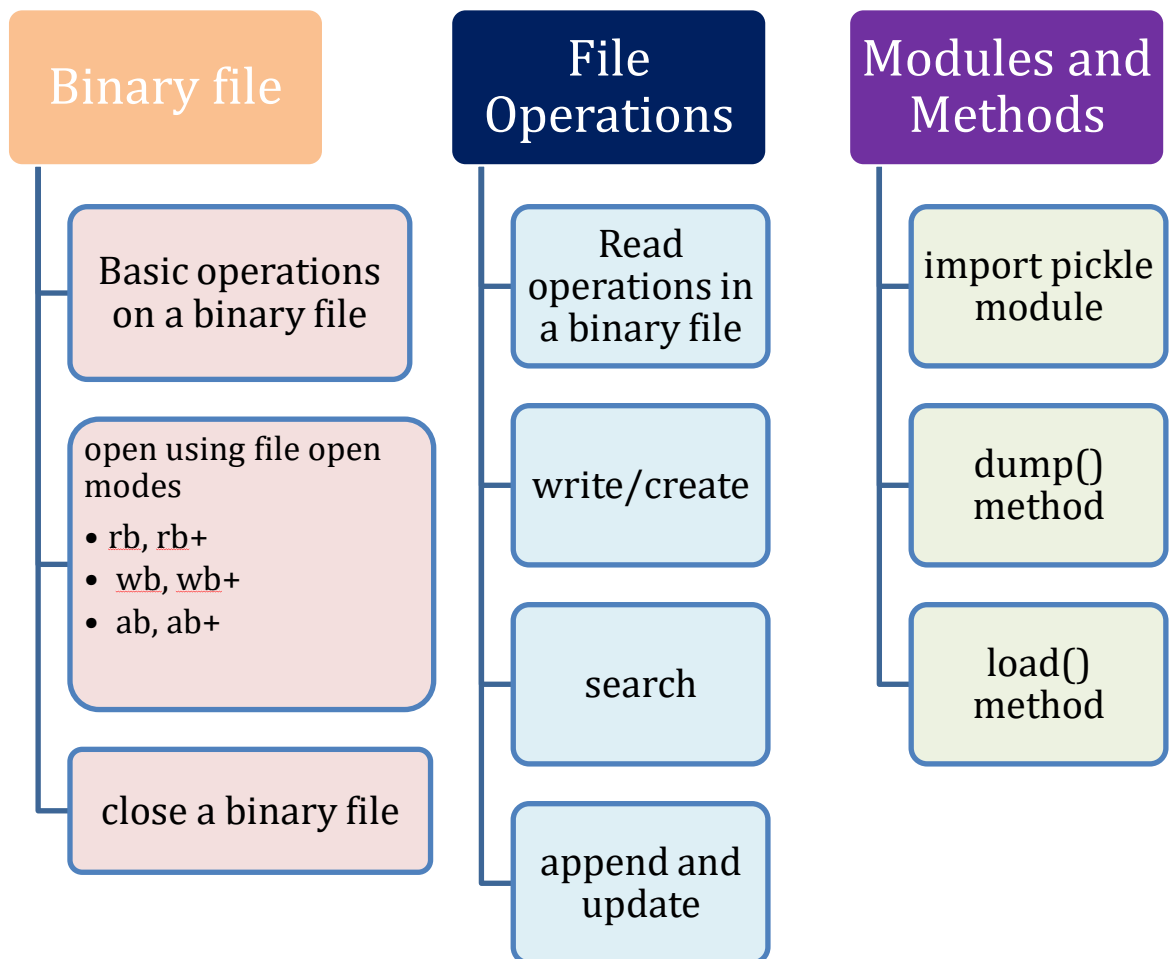
|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 54. | <p>Teacher has developed a program to count how many lines start with "T". To choose correct option for executing program successfully.</p> <pre>def countlines():     file=open("story.txt","r")     data=..... #line1     ..... #line2     for i in data:         if.....: #line3             c=c+1     print('number of lines start with T-&gt;c',.....) # line4     file.close() countlines()</pre> <p>choose correct option for line 1</p> <p>a.file.read()<br/>b.readline()<br/>c.file.readlines()<br/>d.file.readline()</p> <p><b>Correct answer:c</b></p> |
| 55. | <p>choose correct option for line2- initialize the variable for count</p> <p>a.c=0<br/>b.c=c+1<br/>c.both<br/>d.none</p> <p><b>Correct answer:a</b></p>                                                                                                                                                                                                                                                                                                                                                                                                           |
| 56. | <p>choose correct option for line3</p> <p>i=='T'<br/>i[0]=='T'<br/>"i"==T<br/>i[0]==T</p> <p><b>Correct answer:b</b></p>                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 57  | <p>complete the line4</p> <p>a.c<br/>b.count<br/>c.i<br/>d.data</p> <p><b>Correct answer:a</b></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

## Binary Files

- Binary file(used to store binary data such as images, video files, audio files etc.) is a non-text file. It contains data as 1s and 0s(computer readable format).
- Binary files are processed byte by byte.
- There is no delimiter and EOL character in the binary file.
- Binary files are faster in processing and consumes less memory compared to text files.
- Extension of binary files .bin(.dat allowed?)
- pickle module is offering functions(dump and load) to operating binary files.
- File opening mode must attach 'b' to it for operating binary file(Ex: 'rb'- for reading)

Basic operations with binary files:

- a. Reading data from a file
- b. Writing data to a file
- c. Appending data to a file
- d. Deleting a file





## SECTION - A

1. Which of the following statement opens a binary file record.bin in write mode and writes

data from a list lst1 = [1,2,3,4] on the binary file?

**a. with open('record.bin','wb') as myfile:**

**pickle.dump(lst1,myfile)**

b. with open('record.bin','wb') as myfile:

pickle.dump(myfile,lst1)

c. with open('record.bin','wb+') as myfile:

pickle.dump(myfile,lst1)

d. with open('record.bin','ab') as myfile:

pickle.dump(myfile,lst1)

2. Which module is required to use built in function dump()

(a) Math (b)flush **(c)pickle** (d)csv

3.Which of the following function is used to write data in binary mode?

(a)write (b)output **(c)dump** (d)send

4.To read 2 characters from file object f1 command should be

**(a)f1.read(2)** (b)f1.read() (c)f1.readline() (d) f1.readlines()

5.To get byte position from the beginning of file, function used is -

(a)seek **(b)tell** (c)read (d)write

6.The file pointer, used to go to particular position

**(a)seek** (b)tell (c)read (d)write

7.If a file is opened for reading, which of the following statements is not true?

a. The file must exist on the disk on the specified path

b. If the file exists at the specified path, the file is successfully opened.

- c. The file even if at a different location on disk other than the specified path, will get opened
- d. Python gives error if the file does not exist at the specified path

**Answer : c) The file even if at a different location on disk other than the specified path, will get opened.**

8.To read 24 characters from a file object infi, we use

- a. Infi.read() b. infi.read(24)
- c. Infi.readline() d. infi.readlines

**Answer: b) infi.read(24)**

9.The readlines() method returns\_\_\_\_\_

- a. a str b. a list of integers
- c. a list of single characters d. a list of lines

**Answer : d) a list of lines.**

10.Which of the following is not a valid mode to open a file.

- a. ab b. rw
- c. wb d.w+

**Answer : b) rw**

11.Which of the following functions do you use to write data in the binary format?

- a. Write() b. output()
- c. dump() d. send()

**Answer : c) dump().**

12.Which of the following commands can be used to read the entire contents of a file as a string using the file object <tmpfile>?

- a. tmpfile.read(n)
- b. tmpfile.read()
- c. tmpfile.readline()

d. `tmpfile.readlines()`

**Answer : b. `tmpfile.read()`**

13. Which of the following command is used to open a file "c:\temp.txt" for writing in binary format only?

a. `outfile = open("c:\temp.txt", "w")`

b. `outfile = open("c:\\temp.txt", "wb")`

c. `outfile = open("c:\temp.txt", "w+")`

d. `outfile = open("c:\\temp.txt", "wb+")`

**Answer : b. `outfile = open("c:\\temp.txt", "wb")`**

14. \_\_\_\_\_ is the process of converting Python object hierarchy into a byte stream so that it can be written into a file.

a) Pickling

b) Unpickling

c) Dumping

d) Loading

**Answer: a**

15. \_\_\_\_\_ is the process of reading from a binary file

a) Pickling

b) Unpickling

c) Dumping

d) Loading

**Answer: b**

16. \_\_\_\_\_ of pickle module will unpickle the data coming from the binary file.

a) `load()`

b) `dump()`

c) writer()

d) insert()

**Answer : a**

17. \_\_\_\_\_ of pickle module will pickle the data in the binary file.

a) load()

b) dump()

c) writer()

d) insert()

**Answer : b**

18. \_\_\_\_\_ will return the current position of file pointer in the file

a) seek()

b) search()

c) tell()

d) print()

**Answer : c**

19. \_\_\_\_\_ places the file pointer at the specified position in the open file.

a) seek()

b) search()

c) tell()

d) print()

**Answer : a**

20. `F.seek(20,0)` will move the file pointer 20 bytes in forward direction from beginning of file. State True or False

a) True

b) False

**Answer : a**

21. `F1.seek(-5,1)` will move the file pointer 5 bytes backwards from end of file.

State True or False

- a) True
- b) False

**Answer : b**

22. Syntax of seek function in Python is `myfile.seek(offset, reference_point)` where `myfile` is the file object. What is the default value of `reference_point`?

- a) 0
- b) 1
- c) 2
- d) 3

**Answer : a**

23. Which of the following statements is true?

- a) pickling creates an object from a sequence of bytes
- b) pickling is used for object serialization
- c) pickling is used for object deserialization
- d) pickling is used to manage all types of files in Python

**Answer : b**

24. “wb” mode in file handling also can be written as

- (a) `w*b`
- (b) `w/b`
- (c) `w+b`
- (d) none of these

25. What is unpickling?

- (a) It is used for object serialization
- (b) **It is used for object deserialization**
- (c) None of the mentioned
- (d) All of the mentioned

26. What is the extension of binary file

- (a).dat      (b).txt      (c).**bin**      (d).file

27. Identify the invalid mode from the following.

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

28. Which value will be shown if you try to open a binary file using a text

- (a) Default value      (b) Advance Value  
(c) **Garbage value**      (d) Parameter Value

29. Binary file is stored in terms of

- (a) bit      (b) **bytes**      (c) nibble      (d) mnemonics

30. Mode used for reading from pickle file is

- (a) **rb**      (b) r      (c) br      (d) read

31. When file is opened as “with open” then file \_\_\_\_\_ automatically.

- (a) reads      (b) writes      (c) **closes**      (d) opens

32. Which of the following commands is used to open the file “STUDENT.DAT” for writing only in binary format?

- (a) **F= open(“STUDENT.DAT”,’wb’)**  
(b) F= open(“STUDENT.DAT”,’w’)  
(c) F= open(“STUDENT.DAT”,’wb+’)  
(d) F= open(“STUDENT.DAT”,’w+’)

33. Which of the following commands is used to write the list L into the binary file, STUDENT.DAT?

- (a) pickle.write(L,f)  
(b) pickle.write(f, L)  
(c) **pickle.dump(L,F)**  
(d) f=pickle.dump(L)

34. Which of the following commands is used to read each record from the binary file STUDENT.DAT?

(a) **R = pickle.load(F)**

(b) pickle.read(r,f)

(c) r= pickle.read(f)

(d) pickle.load(r,f)

35. 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+

36. Which of the following statement(s) are correct regarding the file access modes?

(a) **'r+' opens a file for both reading and writing. File object points to its beginning.**

(b) 'w+' opens a file for both writing and reading. Adds at the end of the existing file if it exists and creates a new one if it does not exist.

(c) 'wb' opens a file for reading and writing in binary format. Overwrites the file if it exists and creates a new one if it does not exist.

(d) 'a' opens a file for appending. The file pointer is at the start of the file if the file exists.

37. The syntax for dump () method

1. **dump(data\_object, file\_object)**

2. dump[data\_object, file\_object]

3. dump[file\_object]

4. dump[data\_object]

38. Binary files represent the actual content

(a) image (b) audio (c) video (d) **All of these**

39. The process by which a pythonobject is converted to a byte stream is called as

(a) Unpickling (b) loading (c) **pickling** (d) Dumping

40. Which of the following statements correctly explain the function of seek() method?

(a) tells the current position within the file.

(b) determines if you can move the file position or not.

(c) indicates that the next read or write occurs from that position in a file.

(d) moves the current file position to a given specified position

41. What is 'f' in the following statement?

```
f=open("Data.txt" , "r")
```

(a) File Name (b) **File Handle** (c) Mode of file (d) File Handling

## SECTION - B

1. The following code has some errors. Identify the error line(s)

```
import pickle
```

```
f = open ('example.dat', rb)
```

```
f.dump([2,3,4])
```

```
f.close()
```

(a) Line 1 and 2 have the errors

(b) Lines 3 and 4 have errors

(c) Line 2 and 3 have the errors

(d) Lines 2 has error

2. Rahul is trying to read data from a binary file test.bin and store it into List L. Consider the following code written by him.

```
import pickle as p
```

```
with open('test', 'rb') as f:
```

```
_____ #statement 1
```

Identify the missing code in Statement 1.

(a) f=p.load(L)      (b) p=p.load(L)      (c) L=p.load(f)      (d) f=L.load(p)

3. What is the output produced by the following code:



```

import pickle

score =[10,20,30,40,50,60]

with open('test.dat','wb') as fout:

    for ix in score:

        pickle.dump(ix,fout)

    size=0

    with open('test.dat','rb') as fin:

        num=pickle.load(fin)

        size+=num

        num=pickle.load(fin)

        print(size,end=',')

        num=pickle.load(fin)

        size+=num

        print(size)

```

- (a) 10, 50    (b) 10, 40    (c). 20, 50    (d). 10, 60

4. What will be displayed by the following code?

```

import pickle

colors=['red','green','blue','grey','pink']

with open('test.dat','wb') as fout:

    pickle.dump(colors,fout)

    with open('test.dat','rb') as fin:

        lst=pickle.load(fin)

```

```
print(lst[1:5:2])
```

- (a) ['green', 'blue'] (b) ['red', 'blue'] (c) ['green', 'grey'] (d) ['red', 'green']

5. The binary file STOCK.DAT is containing following fields

SName and Price

Function Billing() is created in Python to read each record of a pickled file STOCK.DAT, and display the Total Price of all the records in the file.

```
import pickle

def Billing():

    file=open('STOCK.DAT','rb')

    IRec=pickle.load(file) #To read the object from file

    Totprice=0

    for I in IRec:

        _____

    print (Totprice)

    file.close()
```

Choose appropriate statement to calculate the total price from the following:

- (a) Totprice=I[1] (b) Totprice=+I[1]  
(c) Totprice+=I[1] (d) Totprice=I+Totprice

6. When function SPLDOCS() is executed, it display all the content from a pickled file DOCS.DAT, where Specialisation of DOCTORS is

“CARDIOLOGY”.

```
def Disp(s):
    print (s.Name,"#",s.Specialisation)

def SPLDOCS():
    D=DOCTORS()

    file=open('DOCS.DAT','rb')

    try:

        while True:

            D=pickle.load(file)

            if D.Specialisation == 'CARDIOLOGY':

                _____

    except EOFError:

        pass

    file.close()
```

To fill the missing statement 1, choose the statement to call function Disp

- (a) D[Disp()]                      (b) D.Disp()                      (c) Disp(D)                      (d) Disp[D]

7. Sunil is writing a program to search for the details a particular student. He has got difficulties in completing the code. Help him to complete it.

\_\_\_\_\_ open('student',rb) as \_\_\_\_\_:

Complete the open() statement

(a) filehandle, filename    (b). filename, filehandle

(c) with, filehandle                      (d) with,filename

8. Aman wants to read pickled file ELECTION.DAT and displays every such Name whose Count is less than 10 but facing some problems to give the condition.

The File contains the following fields :Name,Count

```
import pickle
```

```
def cLowCount():
```

```
file=open('ELECTION.DAT','rb')
```

```
ERec=pickle.load(file) #To read the object from file
```

```
for E in ERec:
```

```
    if _____: # statement 1
```

```
        print (E[0] )
```

```
file.close()
```

Choose the correct option which satisfied the condition.

(a) E<10    (b) E[2]<10    (c) E[1]<10                      (d) E[0]<10

9. Observe the below code and write which of the following is converting the list into bytes is

```
list1 = [23, 43, 55, 10, 90]
```

```
newFile = open("binaryFile", "wb")
```

```
newFile.write(bytearray(list1))
```

```
newFile.close()
```

(a) byte array() (b) newFile (c) write                      (d) wb

10. Raghav is trying to write an object `obj1 = (1,2,3,4,5)` on a binary file “test.bin”. Consider the following code written by him.

```
import pickle

obj1 = (1,2,3,4,5)

myfile = open("test.bin", 'wb')

pickle. #Statement 1

myfile.close()
```

Identify the missing code in Statement 1.

- (a) `dump(myfile,obj1)`
- (b) `dump(obj1, myfile)`
- (c) `write(obj1,myfile)`
- (d) `load(myfile,obj1)`

11. . Raghav is trying to write a tuple `tup1 = (1,2,3,4,5)` on a binary file test.bin. Consider the

following code written by him.

```
import pickle

tup1 = (1,2,3,4,5)

myfile = open("test.bin", 'wb')

pickle._____ #Statement 1

myfile.close()
```

Identify the missing code in Statement 1.

a. dump(myfile,tup1)

b. dump(tup1, myfile)

c. write(tup1,myfile)

d. load(myfile,tup1)

12. A binary file employee.dat has following data

| Empno | empname       | Salary |
|-------|---------------|--------|
| 101   | Anuj          | 50000  |
| 102   | Arijita       | 40000  |
| 103   | Hanika        | 30000  |
| 104   | Firoz         | 60000  |
| 105   | Vijayalakshmi | 40000  |

```
def display(eno):
    f=open("employee.dat","rb")
    totSum=0
    try:
        while True:
            R=pickle.load(f)
            if R[0]==eno:
                _____ #Line1
                totSum=totSum+R[2]
    except:
        f.close()
    print(totSum)
```

When the above mentioned function, display (103) is executed, the output displayed is

190000.

Write appropriate jump statement from the following to obtain the above output.

- a. jump
- b. break
- c. continue
- d. return

## SECTION - C

1. Arun, during Practical Examination of Computer Science, has been assigned an incomplete search() function to search in a pickled file student.dat. The File student.dat is created by his Teacher and the following information is known about the file.

- File contains details of students in [roll\_no,name,marks] format.
- File contains details of 10 students (i.e. from roll\_no 1 to 10) and separate list of each student is written in the binary file using dump().

Arun has been assigned the task to complete the code and print details of roll number1.

```
def search(r):
```

```
    f = open("student.dat",____) #Statement-1
```

```
    ____: #Statement-2
```

```
while True:
```

```
    rec = pickle.____ #Statement-3
```

```
if(____): #Statement-4
```

```
    print(rec)
```

```
except:
```

```
pass
```

```
____ #Statement-5
```

(i) In which mode Arun should open the file in Statement-1?

- (a) r                      (b) r+                      (c) rb                      (d) wb

(ii) Identify the suitable code to be used at blank space in line marked as Statement2

- (a) if(rec[0]==1)      (b) for i in range(10)                      (c) try                      (d) pass

(iii) Identify the function (with argument), to be used at blank space in line marked as Statement-3.

- (a) load()      (b) load(student.dat)                      (c) load(f) (d) load(fin)

(iv) What will be the suitable code for blank space in line marked as Statement-4.

- (a) rec[0]=r (b) rec[1]==r                      (c) rec[2]=r (d) rec[0]==r

(v) Which statement Arun should use at blank space in line marked as Statement4 to close the file.

- (a) file.close()                      (b) close(file)                      (c) f.close()                      (d) close()

2. Arun, during Practical Examination of Computer Science, has been assigned an incomplete search() function to search a record which is passed as parameter in a pickled file student.dat. The File student.dat is



created by his Teacher and the following information is known about the file.

- (a) File contains details of students in [roll\_no,name,marks] format.
- (b) File contains details of 10 students (i.e. from roll\_no 1 to 10) and separate list of each student is written in the binary file using dump().

Arun has been assigned the task to complete the code and print details of roll number1.

```
def search(r):
```

```
    f = open("student.dat",____)                #Statement-1
```

```
        ____:                                    #Statement-2
```

```
        while True:
```

```
            rec = pickle.____                    #Statement-3
```

```
            if(____):                             #Statement-4
```

```
                _____                      #Statement-5
```

```
        except:
```

```
            pass
```

```
    _____                                    #Statement-6
```

i. In which mode Arun should open the file in Statement-1?

- (a) r              (b) r+              (c) rb              (d) wb

ii. Identify the suitable code to be used at blank space in line marked as Statement2.

- (a) Try              (b) while True              (c) try              (d) pass

iii. Identify the function (with argument), to be used at blank space in line marked as Statement-3.

- (a) load()      (b) f.load()      (c) load(f)      (d) f(load)

v. What will be the suitable code for blank space in line marked as Statement-4.

a) `rec(0)==r` (b) `rec[1]==r` (c) `rec[1]==r` (d) `rec[0]==r`

v. What will be the suitable code for blank space in line marked as Statement-5 to show record?

(a) `show(rec)` (b) `print(rec)` (c) `print.rec()` (d) `rec(print)`

vi. Which statement Arun should use at blank space in line marked as Statement5 to close the file?

(a) `file.close()` (b) `close(file)` (c) `f.close()` (d) `close()`

3. Ms. Suman 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 `suman_list.dat`, and file object as `f`.

i ) Which of the following can be the correct statement for her?

a) `f = open(„suman_list“, „wb“); pickle.dump(l1,f)`

b) `f = open(„suman_list“, „rb“); l1=pickle.dump(f)`

c) `f = open(„suman_list“, „wb“); pickle.load(l1,f)`

d) `f = open(„suman_list“, „rb“); l1=pickle.load(f)`

Correct Answer : a) `f = open(„suman_list“, „wb“); pickle.dump(l1,f)`

ii ) Which option will be correct for reading file for suman?

a) `f = open(„suman_list“, „rb“)`

b) `f = open(„suman_list“, „r“)`

c) `f = open(„suman_list“, „r+“)`

d) `f = open(„suman_list“, „ab“)`

Correct Answer : a) `f = open(„suman_list“,“rb“)`

iii ) In which of the file mode existing data will be intact in binary file?

- a) a
- b) ab
- c) w
- d) wb

Correct Answer : b) ab

iv ) Which one of the following is correct statement?

- a) `import – pickle`
- b) `pickle import`
- c) `import pickle`
- d) All of the above

Correct Answer : c ) `import pickle`

v) What are the binary files used for?

- a. It is used to store data in the form of bytes.
- b. To store data
- c. To look folder good
- d. None of these

Correct Answer : a) It is used to store data in the form of bytes

4.Ms. Sejal is working on the `sports.dat` file but she is confused about how to complete the code to read the data from the binary file. Suggest a suitable line for her to fulfil her.

```

_____ # Statement 1

def sports_read ():

    f1 = _____ # Statement 2

    _____ # Statement 3

    print(data)

    f1.close ()

sports.read()

```

i ) Identify the suitable code for blank space in line marked as Statement-1.

- a. pickle import
- b. import pickle
- c. import.pickle
- d. None of these

Correct Answer : b) import pickle

ii ) Identify the suitable code for blank space in line marked as Statement-2.

- a. open("sports.dat","wb")
- b. open("sports.dat","r")
- c. open("sports.dat","rb")
- d. None of these

Correct Answer : c) f1 = open("sports.dat","rb")

iii ) Identify the suitable code for blank space in line marked as Statement-3.

- a. data = pickle.load(f1)

b. data = pickle.dump(f1)

c. data = pickle.load(f)

d. data = pickle.dump(f)

Correct Answer: a) data = pickle.load(f1)

iv ) What is the description of `r+b` in binary mode?

a. read and write

b. write and read

c. read only

d. none of these

Correct Answer : a) read and write

v )Which of the following file modes will not delete the existing data in binary file ?

a. wb

b. w

c. a

d. ab

Correct Answer : d) ab

5.Saritha is trying to add data onto a existing binary file and is facing difficulty in

completing the code.Help her to fill the gaps in the code.

Incomplete Code:

```
import pickle
```

```
print("WORKING WITH BINARY FILES")
```

```
_____ # Statement 1
```

```

recno=1

print ("Enter Records of Employees")

print()

#taking data from user and dumping in the file as list object
while True:

    print("RECORD No.", recno)

    eno=int(input("\tEmployee number : "))

    ename=input("_____") # Statement 2

    ebasic=int(input("\tBasic Salary : "))

    allow=int(input("\tAllowances : "))

    totalsal=ebasic+allow

    print("\tTOTAL SALARY : ", totalsal)

    edata=[eno,ename,ebasic,allow,totsal]

    pickle.dump(edata, bfile) # Statement 3

    ans=input("Do you wish to enter more records (y/n)? ")

    recno=recno+1

    if ans.lower()=='n':

        print("Record entry OVER ")

        print()

        break # retrieving the size of file

    print("Size of binary file (in bytes):",

        bfile.tell())

    bfile.write("\n") # Statement 4

```

i ) To open the file for writing the data in line marked as Statement-1.

- a. `bfile=open("empfile.dat","ab")`
- b. `bfile=open("empfile.dat","a")`
- c. `bfile=open("empfile.dat","wb")`
- d. `bfile=open("empfile.dat","w")`

Correct Answer : a) `bfile=open("empfile.dat","ab")`

ii ) To accept employee name from the user in line marked as Statement-2.

- a. `input("\tEmployee Name : ")`
- b. `input(Employee Name :)`
- c. `input("Employee Name )`
- d. None of these

Correct Answer : a) `input("\tEmployee Name : ")`

iii ) Identify the suitable code for blank space in line marked as Statement-3.

- a. `edata,bfile()`
- b. `edata,bfile`
- c. `data,bfile`
- d. `edata,file`

Correct Answer: b) `edata,bfile`

iv ) Identify the suitable code for blank space in line marked as Statement-4.

- a. `bfile.close()`

- b. bfile.close
- c. file.close()
- d. none of these

Correct Answer : a) bfile.close()

v ) Which of the following is the correct syntax to read from a file using load function ?

- A. pickle.load(<filehandle>)
- B. <object> - load.pickle(<filehandle>)
- C. <object> - pickle.load(<filehandle>)
- D. All of the above

Correct Answer : c) <object> - pickle.load(<filehandle>)

6.A Binary file Stock.dat has a structure [pno,pname,qty,price].A user defined function Createfile() to input data for 3 records and add to stock.dat .There are some blanks help in filling the gaps in the code:

Incomplete Code :

Import \_\_\_\_\_ # Statement 1

def createfile():

File=open("d:\\Stock.dat", "\_\_\_\_\_") #Statement 2

pno=input("Enter product no:")

pname= input("Enter product name:")

qty= input("Enter product quantity:")

price= input("Enter product price:")

record=[pno,pname,qty,price]

\_\_\_\_\_ # Statement 3



```
Print("Record inserted")
```

```
File.close()
```

```
Createfile()
```

i ) Identify the suitable code for blank space in line marked as Statement-1.

- a. csv
- b. CSV
- c. pickle
- d. PICKLE

Correct Answer : c) pickle

ii ) Identify the suitable code for blank space in line marked as Statement-2.

- a. wb
- b. ab
- c. w
- d. a

Correct Answer : b) ab

iii ) select correct statement to write data into file for Statement-3.

- a. pickle.dump(record,file)
- b. pickle.dump(record)
- c. pickle.dump(file,record)
- d. pickle.load(record,file)

Correct Answer: a) pickle.dump(record,file)

iv ) Which method is used for object deserialization ?

- A. Pickling
- B. Unpickling
- C. All of the above
- D. None of the above

Correct Answer : b) Unpickling

V )What is the last action that must be performed on a file? \*

- a. save
- b. close
- c. end
- d. write

Correct Answer : b) close

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

```

_____ pickle # line1

def countrec():

fobj=open("_____", "rb") # line2

num = 0

try:

while _____: # line3

rec=pickle.load(fobj)

```

```
if rec[2]>75:  
    num = num + 1  
    print(rec[0],rec[1],rec[2])  
except:  
    fobj.close()  
return num
```

i ) Identify the suitable code for blank space in line-1.

- a. import
- b. IMPORT
- c. Import
- d. None of the above

Correct Answer : a) import

ii ) Identify the suitable code for blank space in line 2.

- a. STUDENT.DAT
- b. STUDENTS.DAT
- c. SCHOOL.DAT
- d. None of the above

Correct Answer : a) STUDENT.DAT

iii ) select correct keyword to fill for line-3.

- a. True
- b. False
- c. true
- d. TRUE

Correct Answer: a) True

6. Ms. Anitha is unable to understand what can be the output of the following code. Help her in getting the output.

```
import pickle
```

```
L=[20,40,50]
```

```
f=open("list.dat","wb")
```

```
Pickle.dump(l,f)
```

```
Print("Data added successfully")
```

```
f.close()
```

```
f=open("list.dat","rb")
```

```
data=pickle.load(f)
```

```
f.close()
```

```
print(data)
```

a. Data added successfully

[20,40,50]

b. [20,30,50]

Data added successfully

c. [20,30,50]

d. No output

Correct Answer: a) Data added successfully

[20,40,50]

8. A binary file "salary.DAT" has structure [teacherid, teacher name, salary]. Complete the code in the blanks so that it would read contents of

the file “salary.DAT” and display the details of those teachers whose salary is above 20000.

```
import pickle

_____ # line1

try:

print("tr id\t tr Name\t tr Sal")

while True:

rec=_____.load(fobj) #line2

if rec[2]>_____: #line3

print(rec[0],"\t\t",rec[1],"\t\t",rec[2])

except:

_____.close() #line 4
```

i ) To open the file for writing the data in line marked as line-1.

- a. fobj=open("salary.dat","rb")
- b. fobj=open("salary.dat","r")
- c. fobj=open("salary.dat","r+")
- d. fobj=open("data.dat","rb")

Correct Answer : a) fobj=open("data.dat","rb")

ii ) The module used in line2

- a. PICKLE
- b. pickling
- c. pickle
- d. None of these

Correct Answer : c) pickle

iii ) Identify the salary to be checked in the code marked as line-3.

- a. 50000
- b. 20000
- c. 24000
- d. 10000

Correct Answer: b) 20000

iv ) Which of the following File Modes creates a new file, if the file does not exist?

(choose one/more)

- a. „r“
- b. „bw“
- c. „w“
- d. „a“

Correct Answer : c) w

v ) What is true about Binary files

- a. They are not human readable
- b. the file extension is .dat
- c. the file stores same format as held in memory.
- d. All of the above

Correct Answer : d) All of the above

8.Mr.Rohan wants to modify salary of employee having a structure[eid,ename

,salary],but unable to fill the gaps in the code. Help him to complete the code

Import pickle

f = open('d:/student.dat','rb')

reclst = []

r=\_\_\_\_\_ # line 1 code to ask employee id

m=int(input("enter correct salary"))

while True:

try:

rec = pickle.load(f)

reclst.append(rec) #line2 statement to add items in list at the end one by

one

except EOFError:

break

f.close()

for i in range (len(reclst)):

if reclst[i]['eid']==r:

reclst[i]['salary'] = m

f = open('d:/student.dat','\_\_\_\_') #line 3 mode to be used to copy the data

for x in reclst:

pickle.dump(x,f)

f.close()

i ) Identify the code in line1.

- a. `int(input("Enter employee id"))`
- b. `int("Enter employee id")`
- c. `int(INPUT("Enter employee id"))`
- d. None of the above

Correct Answer : a) `int(input("Enter employee id"))`

ii ) The module used in line2

- a. PICKLE
- b. pickling
- c. pickle
- d. None of these

Correct Answer : c) pickle

iii ) Fill in the code marked as line-3.

- a. w
- b. wb
- c. r
- d. Rb

Correct Answer: b) wb

9.A binary file `sports.dat` contains information in the following structure:(  
Event, Participant ) A code is shown below which is incomplete that  
would read contents from the `sports.dat` and creates a file named



Athletic.dat copying only those records from sports.dat where the event name is "Athletics".

```
import pickle
```

```
ath ( f1 , f2 ) :
```

```
    l = pickle.load ( f1)
```

```
    for t in l :
```

```
        if ( t [ 0 ] == " _____ " ) : #line 1
```

```
            pickle. _____ ( t , f2 ) #line 2
```

```
f1 = open ( " sports.dat " , " rb " )
```

```
f2 = open ( " athletics.dat " , " wb " )
```

```
f.close()
```

```
f1.close()
```

i ) Identify the code in line1.

- a. Athletics
- b. Sports
- c. Games
- d. None of the above

Correct Answer : a) Athletics

ii ) The function to copy the data into other binary file2

- a. DUMP
- b. close
- c. dump

d. None of these

Correct Answer : c) dump

iii ) Information stored on a storage device with a specific name is called as \_\_\_\_\_.

a. array

b. dictionary

c. file

d. Tuple

Correct Answer: c) file

iv) Which of the following is not a valid mode to open a file?

a . ab

b. rw

c . r+

D.w+

Correct Answer: b) rw

10.A function searchprod( pc) in python is created to display the record of a particular product from a file product.dat whose code is passed as an argument. Structure of product contains the following elements [product code , product price].There is some problem in completing the code,help to finish the code:

```
f = _____('d:/product.dat','rb') #line1
```

```
flag = False
```

```
pc=input("Enter product code to be searched")
```

```
while True:
```

```
    try:
```

```
rec = pickle.load(f)

if rec['pcode'] == ____: #line2
    print('Product code:',rec['pcode'])
    print('Price:',rec['price'])
    flag = True
except EOFError:
    break

if flag == False:
    print('No Records found')

f.close()
```

i ) Identify the method in line1.

- a. close
- b. open
- c. OPEN
- d. None of the above

Correct Answer : b) open

ii ) The variable used to accept product code entered by the user for the line2

- a. pcode
- b. pc
- c. code
- d. None of these

Correct Answer : b) pc

11. Archit wants to create and display a Binary file named "Myfile.DAT". Complete the missing code to open, create and display the file.

```
import _____ #Line1

double=[]

for i in range(1,11):

    double.append(2*i)

fo=_____ #Line2

pickle._____ #Line 3

fo.close()

fin=_____ #Line4

result=_____ #Line 5

fin.close()

print(" The content of file :", result)
```

1. Name the module he should import in Line 1.

- a) csv
- b) pickle
- c) binary
- d) bin

Answer : b

2. Fill in the blank in Line 2 to open the file for writing the contents of the file.

- a) open("Myfile.dat","w")
- b) open("Myfile.dat","r")
- c) open("Myfile.dat","wb")

d) `open("Myfile.dat","rb")`

Answer : c

3. Fill in the blank in Line 3 with the function to write entire contents to file.

- a) `load(double,fo)`
- b) `dump(double,fo)`
- c) `writer(double)`
- d) `insert(double,fo)`

Answer : b

4. Fill in the blank in Line 4 to open the file for displaying contents of file.

- a) `open("Myfile.dat","w")`
- b) `open("Myfile.dat","r")`
- c) `open("Myfile.dat","wb")`
- d) `open("Myfile.dat","rb")`

Answer : d

5. Fill in the blank in Line 5 read the contents of the file.

- a) `pickle.read(fin)`
- b) `pickle.readline(fin)`
- c) `pickle.readlines(fin)`
- d) `pickle.load(fin)`

Answer: d

12. Rohith has been given the following incomplete code for entering his details (Name, contact number and address) to a file "Personal.DAT" and

display the contents. Complete the missing code to open, create and display the file.

```
import _____ #Line1

mydata=[]

name=input("Enter Name:")

contactno=int(input("Enter contact number:"))

address=input("Enter address:")

mydata=[name,contactno,address]

f1=_____ #Line2

pickle._____ #Line 3

f1.close()

f2=_____ #Line4

result=_____ #Line 5

f2.close()

print(" The content of file :", result)
```

1. Name the module he should import in Line 1.

- a) csv
- b) pickle
- c) binary
- d) bin

Answer : b

2. Fill in the blank in Line 2 to open the file for writing the contents of the file.

- a) open("Personal.dat","w")

- b) `open("Personal.dat","r")`
- c) `open("Personal.dat","wb")`
- d) `open("Personal.dat","rb")`

Answer : c

3. Fill in the blank in Line 3 with the function to write entire contents to file.

- a) `load(mydata,f1)`
- b) `dump(mydata,f1)`
- c) `writer(mydata)`
- d) `insert(mydata,f1)`

Answer : b

4. Fill in the blank in Line 4 to open the file for displaying contents of file.

- a) `open("Personal.dat","w")`
- b) `open("Personal.dat","r")`
- c) `open("Personal.dat","wb")`
- d) `open("Personal.dat","rb")`

Answer : d

5. Fill in the blank in Line 5 read the contents of the file.

- a) `pickle.read(f2)`
- b) `pickle.readline(f2)`
- c) `pickle.readlines(f2)`
- d) `pickle.load(f2)`

Answer: d

13. You are provided with some incomplete code for entering student's details (Rollno,

Name and marks) to a file "Student.DAT" and display the contents. Complete the

missing code to open, create and display the file.

```
import _____ #Line1
data=[]
rollno=int(input("Enter Roll number:"))
name=input("Enter Name:")
marks=int(input("Enter mark:"))
data=[rollno,name,marks]
fout= open("Student.dat","wb")
pickle._____ #Line 2
fout.close()
fin=_____ #Line3
output=_____ #Line 4
fin.close()
if _____>=33: #Line5
    print(output[1]," passed")
else:
    print(output[1]," failed")
```

1. Name the module to import in Line 1.

- a) csv
- b) pickle



- c) binary
- d) bin

Answer : b

2. Fill in the blank in Line 2 with the function to write entire contents to file.

- a) load(data,fout)
- b) dump(data,fout)
- c) writer(data)
- d) insert(data,fout)

Answer : b

3. Fill in the blank in Line 3 to open the file for displaying contents of file.

- a) open("Student.dat","w")
- b) open("Student.dat","r")
- c) open("Student.dat","wb")
- d) open("Student.dat","rb")

Answer : d

4. Fill in the blank in Line 4 read the contents of the file.

- a) pickle.read(fin)
- b) pickle.readline(fin)
- c) pickle.readlines(fin)
- d) pickle.load(fin)

Answer: d

5 . Fill in the blank in Line 5 to display he status of students(passed/failed) based on their mark.

- a) Output[0]
- b) Output[1]
- c) Output[2]
- d) output

Answer : c

14. Ritesh wants to perform the following binary file operations, as a part of his assignment, with the help of two user defined functions/modules:

- a. AddEmp() to create a binary file called Employee.DAT containing employee information – employee number, name and salary.
- b. ViewEmp() to display the name and salary of employees who are getting Rs.50000 above as salary. The function should also display the average salary. Help him in filling incomplete code.

```
import pickle
```

```
def AddEmp():
```

```
    _____#Line1 to open the binary file to write data
```

```
    while True:
```

```
        Empno = int(input("Employee number: "))
```

```
        Name = input("Name : ")
```

```
        Salary = int(input("Enter Salary :"))
```

```
        L = [Empno, Name, Salary]
```

```
    #Line2 to write the list L into the file
```

```
Choice = input("enter more (y/n): ")
```

```
if Choice in "nN":
```

```
    break
```

```
F.close()
```

```
def ViewEmp():
```

```
    Total=0
```

```
    Countrec=0
```

```
    C50K=0
```

```
    F=open("Employee.DAT","rb")
```

```
    while True:
```

```
        try:
```

```
            _____#Line3 to read from the file
```

```
            Countrec+=1
```

```
            Total+=R[2]
```

```
            if _____ > 50000: #Line4
```

```
                print(R[1],"has salary ",R[2])
```

```
                C50K+=1
```

```
        except:
```

```
            break
```

```
    if C50K==0:
```

```
        print("No employee with salary more than 50000")
```

```
    average=_____ #Line5 to find averagesalary
```

```
print("average Salary = ",average)
```

```
AddEmp()
```

```
ViewEmp()
```

1. Write statement #Line1, to open the file "Employee.DAT" for writing only in binary format?

- a) F= open("Employee.DAT",'wb')
- b) F= open("Employee.DAT",'w')
- c) F= open("Employee.DAT",'wb+')
- d) F= open("Employee.DAT",'w+')

Answer : a

2. Write statement, #Line2, to write the list L into the binary file, Employee.DAT?

- a) pickle.write(L,f)
- b) pickle.write(f, L)
- c) pickle.dump(L,F)
- d) f=pickle.dump(L)

Answer : c

3. Write statement, #Line3, to read each record from the binary file Employee.DAT?

- a) R = pickle.load(F)
- b) pickle.read(r,f)
- c) r= pickle.read(f)
- d) pickle.load(r,f)

Answer : a

4. Write statement , #Line4, to find employees who are getting salary more than Rs.50000.

- a) R[0]
- b) R[1]
- c) R[2]
- d) R[3]

Answer : c

5. Write statement , #Line5, to find average salary of employees

- a) Total/countrec
- b) Total/C50K
- c) Total/Countrec
- d) average(Total)

Answer : c

15.Anand, a software developer, is asked to help a librarian to find some details of books in his library. The book information is stored in a binary file Books.DAT. Create two user defined functions/modules:

- a. AddBook() to create a binary file called Books.DAT containing Book information – Book name, Author and Price.
- b. ViewBook() to display the name, Author and price of books which are more than Rs.350 in price. The function should also display the average price of books.

Try to fill the incomplete code to get required information for librarian.

```
import pickle
```

```
def AddBook():
```

\_\_\_\_\_#Line1 to open the binary file to write data

while True:

    Name = input("Book Name : ")

    Author= input("Author Name :")

    Price = int(input("Enter Book Price:"))

    Lst= [Name, Author, Price]

    #Line2 to write the list Lst into the file

    Choice = input("enter more (y/n): ")

    if Choice in "nN":

        break

Fp.close()

def ViewBook():

    Total=0

    Count=0

    C=0

    F=open(\_\_\_\_\_) #Line3 to open file for reading

    while True:

        try:

            \_\_\_\_\_#Line4 to read from the file

            Count+=1

            Total+=Row[2]

            if \_\_\_\_\_ > 350: #Line5

            print("Price of ",Row[1],"=

```
        ",Row[2])
```

```
        C+=1
```

```
    except:
```

```
        break
```

```
if C==0:
```

```
    print("All books are having price less than 350")
```

```
    avgprice=Total/Count
```

```
    print("Average Price = ",avgprice)
```

```
    F.close()
```

```
AddBook()
```

```
ViewBook()
```

1. Fill in the blank in Line1, to open the file “Books.DAT” for writing in binary format?

- a) Fp= open("Books.DAT",'w')
- b) Fp= open("Books.DAT",'wb')
- c) Fp= open("Books.DAT",'wb+')
- d) Fp= open("Books.DAT",'w+')

Answer : b

2. Fill in the blank in Line2, to write the list Lst into the binary file, Books.DAT?

- a) pickle.write(Lst,fp)
- b) pickle.write(fp, Lst)

c) `pickle.dump(Lst,Fp)`

d) `fp=pickle.dump(Lst)`

Answer : c

3. Fill in the blank in Line3, to open the file “Books.DAT” for reading

a) `F= open("Books.DAT",'r')`

b) `F= open("Books.DAT",'r+')`

c) `F= open("Books.DAT",'wb+')`

d) `F= open("Books.DAT",'rb')`

Answer : d

4. Fill in the blank in Line4, to read data from file.

a) `Row=pickle.load(Fp)`

b) `Row=pickle.read(Fp)`

c) `Row=pickle.read(F)`

d) `Row=pickle.load(F)`

Answer : d

5. Fill in the blank in Line5 with suitable expression:

a) `Row[0]`

b) `Row[1]`

c) `Row[2]`

d) `Row[3]`

Answer : c



## CSV Files

CSV stands for “Comma Separated Values.”

It is the simplest form of storing data in tabular form as plain text.

Each line of the file is a data record.

Each record consists of one or more fields, separated by commas.

Default delimiter of CSV File is comma

Reading and writing CSV files can be done with the help of csv module in Python

### Features:

It is Easier to create

It is the preferred import and export format for databases and spread sheets

It is capable of storing large amount of data

### Module Used:

csv Module

### Functions Used :

open() : opens the csv file

csv.writer() : returns writer object which writes data into csv file

csv.writerow() : writes one row of data onto writer object

csv.writerows() : writes multiple rows of data onto writer object

csv.reader() : returns a reader object which loads data from CSV file into an iterable object

**Multiple Choice Questions:****1. CSV stands for :**

- a. Comma Separated Values
- b. Comma Separated Variables
- c. Comma Stored Values
- d. Comma Stored Variables

**2. The separator character of CSV Files is called a**

- a. EOL
- b. Delimiter
- c. EOF
- d. Default

**3. The default delimiter for a CSV file is :**

- a. Semi colon
- b. Colon
- c. Comma
- d. Hyphen

**4. \_\_\_\_\_ module of Python provides the functionality to read and write tabular data in CSV file.**

- a. pickle
- b. csv
- c. file
- d. ccv

**5. Which function is used to open a csv file ?**

- a. Open()
- b. csv.open()
- c. writer()
- d. csv.writer()

**6. Name the function to read from CSV file.**

- a. read()
- b. csv.reader()
- c. csv.read()
- d. readline()

**7. Which among the following is not a function of csv module?**

- a. reader()
- b. read()
- c. writer()
- d. writerows()

**8. In Python, default newline character is :**

- a. \f
- b. \t
- c. \n
- d. \v

**9. CSV module allows to write multiple rows using \_\_\_\_\_ function.**

- a. writerows( )
- b. . writerow( )
- c. writer( )
- d. None of the above

**10. The opening function of a csv file is similar to the opening of:**

- a. Binary file
- b. Text File
- c. Both of them
- d. None of them

**11. The \_\_\_\_\_ argument of open function is to specify how Python handle the newline characters in csv file**

- a. newline
- b. line
- c. mode
- d. char

**12. Which among the following is an iterable object ?**

- a. writer
- b. file
- c. reader
- d. All of the above

**13. To specify a different delimiter while writing into a csv file, \_\_\_\_\_ argument is used with writer object :**

- a. newline
- b. separator
- c. character
- d. delimiter

**14. Which mode opens the file for exclusive creation, which fails in the case where file already exists**

- a. a
- b. w
- c. x
- d. r

**15. The file mode to open a CSV file for reading as well as writing is \_\_\_\_\_.**

- a) a+
- b) w+
- c) r+
- d) All the above.

**16. Identify the line number which may cause an error:**

```
import csv #Line1
line=[[1,2,3],[4,5,6]]#Line 2
with open("sample.csv","w",newline="") as csvfile: #Line3
    writer=csv.writer(csvfile,delimiter="|") #Line4
    for line in writer: #Line5
        writer.writerow(line)
```

- a. Line1
- b. Line2
- c. Line 4
- d. Line 5

**17. The CSV files are \_\_\_\_ files.**

- a) Plain text file
- b) Binary
- c) Data
- d) Python

**18. The writer() function has how many mandatory parameters?**

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

**19. Which of the following parameter needs to be added with open function to avoid blank row followed by each record in the CSV file?**

- a. quotechar
- b. quoting
- c. newline
- d. skiprow

**20. Ajaikrishna wants to separate the values by a \$ sign. Suggests him a pair of function and parameter to use it.**

- a. open, quotechar
- b. writer, quotechar
- c. open, delimiter
- d. writer, delimiter

### Case Study Based Questions:

**21. Tejalakshmi of Class 12 have written the below code . Observe and fill in the given blanks so that it opens the file “data.csv” and read and print all the records.**

```
import csv
with _____ as f: #1
    r = _____(f) #2
    for row in _____: #3
        print(_____) #4
```

- I. What must be filled in line 1?**
- a. `Open("data.csv","r")`
  - b. `open("data.csv")`
  - c. `"data.csv"`
  - d. `File`
- II. What must be filled in Line 2?**
- a. `csv.reader()`
  - b. `csv.read()`
  - c. `csv.write()`
  - d. `csv.writer()`
- III. What must be filled in line 3?**
- a. `data.csv`
  - b. `f`
  - c. `r`
  - d. `None`
- IV. What must be filled in line 4?**
- a. `data`
  - b. `f`
  - c. `"File"`
  - d. `row`
- V. What is the default data type of data read from this file?**
- a. List
  - b. String
  - c. Tuple
  - d. Integer

**22.Sudev, a student of class 12th, is learning CSV File Module in Python. During examination, he has been assigned an incomplete python code to create a CSV file 'customer.csv' .Help him in completing the code which creates the desired CSV file.**

| Cus_No | Name  | Ph_No      |
|--------|-------|------------|
| 11     | Rohit | 8567843243 |
| 12     | Sonal | 9645342345 |

```

_____ csv #Statement 1
def Create_CSV():
    fw=open("customer.csv","w")
    _____=csv.writer(fw) #Statement 2
    csvwriter.writerow(["Cus_No","Name","Ph_No"])
    n=int(input("Enter total number of Customer"))
    for i in range(n):
        cusno=int(input("Enter Customer no. "))
        Name=input("Enter Name")
        Ph_No=int(input("Enter Phone No. "))
        Rec=[cusno,Name,Ph_No]
        csvwriter.writerow(_____) #Statement 3
    fw.close()

def Display_CSV():
    fr=open(_____, "r") #Statement 4
    cusreader=csv.reader(fr)
    i=0
    for _____ in cusreader: #Statement 5
        if i%2==0:
            print(rec[0],'\t',rec[1],'\t',rec[2])
        else:
            pass
        i+=1
    fr.close()

```

Create\_CSV()  
Display\_CSV(

**I. Identify suitable code for the blank space in line marked as Statement-1.**

- a) include
- b) add
- c) Import
- d) import

**II. Identify the missing code for the blank space in line marked as Statement-2.**

- a) Customer
- b) reader
- c) csvwriter
- d) writer

**III. Identify the argument name for the blank space in line marked as Statement-3?**

- a) Row
- b) Rec
- c) row
- d) rec

**IV. Identify the missing file name for the blank space in line marked as Statement-4?**

- a) customer
- b) customer.csv
- c) customer.txt
- d) customer.dat



V. Identify the object name for the blank space in line marked as Statement-5?

- a) i
- b) Rec
- c) row
- d) rec

**23. Daya of class 12 is writing a program to create a CSV file “empdata.csv” with empid, name & mobile number. Also to search a particular empid and display its record details. He has written the following code. As a programmer, help him to successfully execute the given task.**

```
import _____ #Line1
fields=['empid','name','mobile_no']
rows=[['101','Rohit','8982345659'],['102','Shaurya','8974564589']]
filename="empdata.csv"
f=open(filename,'w',newline=")
csv_w=_____ #Line2
for row in rows:
    csv_w. _____ #Line3
f.close()
f=open(filename,'r')
csv_r=_____ #Line4
ans='y'
while ans=='y':
    found=False
    emplid=input("Enter employee id to search=")
    for row in csv_r:
        if len(row)!=0:
            print(row[0])
            if _____==emplid: #Line5
                print("Name : ",row[1])
                print("Mobile No : ",row[2])
                found=True
                break
    if not found:
        print("Employee id not found")
```

```
ans=input("Do you want to search more? (y)")
```

**I. Choose the module he should import in Line 1.**

- a) math
- b) pickle
- c) csv
- d) random

**II. Choose a code to write the column heading from fields list in Line2.**

- a) writerows(fields)
- b) writerow(field)
- c) writerow(fields)
- d) writerows(fields)

**III. Choose a code to write the row from rows list in Line3.**

- a) writerows(row)
- b) writerow(row)
- c) writerow(rows)
- d) write\_row(row)

**IV. Choose a code for line 4 to read the data from a csv file.**

- a) csv.reader(f)
- b) csv.read(f)
- d) pickle.load(f)
- e) f.read()

**V. Choose the correct variable (list value) to check “emplid” in Line5.**

- a) Row[0]
- b) Rec[0]
- c) row[0]
- d) rec[0]

**24. Viraj is making a software on “Countries and their Capitals” in which various records are to be stored/retrieved in “CAPITAL.CSV” data file. It consists of few records of Countries and their Capitals. He has written the following code in python. As a programmer, you have to help him to successfully execute the program.**

```
import csv
# Function to add a new record in CSV file
def _____(Country,Capital): # Statement-1
    f=open("CAPITAL.CSV","__") # Statement-2
    fwriter=csv.writer(f)
    fwriter.writerow([____]) # Statement-3
    f.close()
def ShowRec(): # Fn. to display all records from CSV file
    with open("CAPITAL.CSV","r") as NF:
        NewReader=csv._____ (NF) # Statement-4
        for rec in NewReader:
            if len(rec)!=0:
                print(rec[0],rec[1])
AddNewRec("INDIA","NEW DELHI")
AddNewRec("CHINA","BEIJING")
ShowRec() # Statement-5
```

**I. Choose the Name of the function in Statement-1.**

- a) AddNewRec

- b) Addnew
- c) Addrec
- d) AddNewRec()

**II. Choose the file mode to be passed to add new records in Statement-2.**

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

**III. Identify the correct variables in Statement-3 to store data to the file.**

- a) country,capital
- b) Country,Capital
- c) Coun,Cap
- d) [Country,Capital]

**IV. Choose the correct option for Statement-4 to read the data from a csv file.**

- a) Reader()
- b) reader()
- c) read
- d) reader

**V. Choose the output which will come after executing Statement-5.**

- a) 'INDIA NEW DELHI'  
'CHINA BEIJING'
- b) 'CHINA' 'BEIJING'
- c) INDIA NEW DELHI

d) None of the above

**25. Rinsha of class 12 is writing a program to create a CSV file “user.csv” which will contain user name and password for some entries. She has written the following code. As a programmer, help her to successfully execute the given task.**

```
import _____ # Line 1

def addCsvFile (UserName, PassWord): # to write / add data into the CSV file
    f = open ('user.csv', '_____') # Line 2
    newFileWriter = csv . writer ( f )
    newFileWriter . writerow ( [ UserName, PassWord])
    f . close ( )
# csv file reading code
def readCsvFile ( ):
    with open ('user.csv', 'r') as newFile:
        newFileReader = csv . _____ (newFile) #Line3
        for row in newFileReader :
            print (row[0], row [1])
    newFile . _____ # Line 4

addCsvFile ("Biplab", "123@456")
addCsvFile ("Arunima", "aru@nima")
addCsvFile ("Poonam", myname@FRD")
readCsvFile ( ) # Line 5
```

**I. Name the module she should import in Line 1.**

- a. CSV
- b. csv
- c. math
- d. File

**II. In which mode, Rinsha should open the file to add data into the file.**

- a. r
- b. a
- c. w
- d. w+

**III. Fill in the blank in Line 3 to read the data from a csv file.**

- a. writer()
- b. reader()
- c. write()
- d. read()

**IV. Fill in the blank in Line 4 to close the file.**

- a. End()
- b. Close()
- c. close()
- d. end()

**26. Consider the following csv file and the code fragment associated with the following csv file :**

| SLNO | BOOKNAME             | PRICE |
|------|----------------------|-------|
| 1    | Pride and Prejudice  | 200   |
| 2    | Gone with the Wind   | 250   |
| 3    | The little prince    | 170   |
| 4    | Anne of Green Gables | 190   |
| 5    | The Giving Tree      | 210   |

```
import csv
f=open("c:\\stud.csv")
fobj=csv.reader(f)
for k in fobj:
    for j in k:
        print(j, end='')
    break
f.close()
```

**I.What will be the output printed by the above code?**

- a. SLNO12345
- b. SLNO
- c. The entire content
- d. Error

**II.What will be the output printed by the above code if the break is replaced with continue?**

- a. SLNO12345
- b. SLNO
- c. The entire content
- d. Error

**III.What will occur if the file stud.csv is not existing in the mentioned path?**

- a. It will create a new one
- b. It will create an error
- c. None of the above
- d. It will cause a system reboot

**IV.Which statement in the above code will help to move to the next record?**

- a. fobj.next()
- b. next(fobj)
- c. fobj.move()
- d. fobj.forward()

**27.Sai Krishna has created the following csv file named item.csv:**

| ITEMNO | NAME      | PRICE |
|--------|-----------|-------|
| 101    | PENCIL    | 5     |
| 102    | PEN       | 10    |
| 103    | NOTE BOOK | 15    |

**He has written the following program for managing the file. Help him to find the answers for the following questions.**

```
import csv
csvfile=open("c:\\item.csv","r")
```

```

csvr=csv.reader(csvfile)
head=next(csvfile)
print(head)#Line1
for i in csvr:
    print(i)#Line2
    break
else:
    print(i)#Line3
k=next(csvfile)
print(k)#Line4
_____#Line5

```

**I. What will be printed by Line1?**

- a. All the records
- b. ITEMNO, NAME, PRICE
- c.item.csv
- d. None of the above

**II. What will be printed by Line2?**

- a. 101,PENCIL,5
- b. ITEMNO, NAME, PRICE
- c.102,PEN,10
- d. 103,NOTEBOOK,15

**III. What will be printed by Line3?**

- a. 103,NOTEBOOK,15
- b. Line 3 will not be executed
- c. Line 3 will be executed , but nothing will be printed
- d. 102,PEN,10

**IV. What will be printed by Line4?**

- a. 101,PENCIL,5
- b. ITEMNO, NAME, PRICE
- c.102,PEN,10
- d. 103,NOTEBOOK,15



**V. What must be written in Line 5?**

- a. F.close()
- b. f.close()
- c. fobj.close()
- d. csvfile.close()

**Assertion and Reasoning**

28.Assertion: **All files must be closed after use.**

Reason :

- A. Function close() is a compulsory part of the syntax
- B. Data will be stored in secondary memory when we close the file

- a. Only A is True
- b. Only B is True
- c. Both A and B are True
- d. Both A and B are False

29.Assertion: **When we create a new file it is better to use the opening mode 'x'**

Reason:

- A. It will avoid overwriting an existing file
- B. It will prevent the risk of data loss

- a. Only A is True
- b. Only B is True
- c. Both A and B are True
- d. Both A and B are False

30. Assertion:

**The new record written in a file opened in a+ mode, will be added at the end.**

Reason :

- A.The file pointer is at the end of the file if the file exists.
- B.The file will be created as a new file

- a. Only A is True
- b. Only B is True
- c. Both A and B are True
- d. Both A and B are False

## ANSWERS

| QNO | OPT | QNO | OPT | QNO | OPT | QNO | OPT |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 1   | a   | 6   | b   | 11  | a   | 16  | d   |
| 2   | b   | 7   | b   | 12  | c   | 17  | a   |
| 3   | c   | 8   | c   | 13  | d   | 18  | a   |
| 4   | b   | 9   | a   | 14  | c   | 19  | c   |
| 5   | b   | 10  | b   | 15  | d   | 20  | d   |

| QNO |     | OPT | QNO |     | OPT | QNO |     | OPT |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 21  | I   | b   | 22  | I   | d   | 23  | I   | c   |
|     | II  | a   |     | II  | c   |     | II  | c   |
|     | III | c   |     | III | b   |     | III | b   |
|     | IV  | d   |     | IV  | b   |     | IV  | a   |
|     | V   | b   |     | V   | d   |     | V   | c   |

| QNO |     | OPT | QNO |     | OPT | QNO |     | OPT |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 24  | I   | d   | 25  | I   | b   | 26  | I   | b   |
|     | II  | d   |     | II  | b   |     | II  | c   |
|     | III | b   |     | III | b   |     | III | b   |
|     | IV  | D   |     | IV  | c   |     | IV  | b   |
|     | V   | b   |     |     |     |     |     |     |

| QNO |     | OPT | QNO | OPT |
|-----|-----|-----|-----|-----|
| 27  | I   | b   | 28  | b   |
|     | II  | a   | 29  | c   |
|     | III | b   | 30  | c   |
|     | IV  | c   |     |     |
|     | V   | b   |     |     |

**Sample Question Paper**  
**Class: XII Session: 2021-22**  
**Computer Science (Code 083)**  
**(Theory: Term-1)**

**Maximum Marks: 35**

**Time Allowed: 90 Minutes**

**General Instructions:**

- The question paper is divided into 3 Sections - A, B and C.
- Section A, consist of 25 Questions (1-25). Attempt any 20 questions.
- Section B, consist of 24 Questions (26-49). Attempt any 20 questions.
- Section C, consist of 6 case study based Questions (50-55). Attempt any 5 questions.
- All questions carry equal marks.

| Q.N. | Section-A                                                                                                                                                                                                                                                                                                                                                                 |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|      | <b>This section consists of 25 Questions (1 to 25). Attempt any 20 questions from this section. Choose the best possible option.</b>                                                                                                                                                                                                                                      |
| 1    | Find the invalid identifier from the following<br>a. none<br>b. address<br>c. Name<br>d. pass                                                                                                                                                                                                                                                                             |
| 2    | Consider a declaration L = (1, 'Python', '3.14').<br>Which of the following represents the data type of L?<br>a. list<br>b. tuple<br>c. dictionary<br>d. string                                                                                                                                                                                                           |
| 3    | Given a Tuple tup1= (10, 20, 30, 40, 50, 60, 70, 80, 90).<br>What will be the output of print (tup1 [3:7:2])?<br>a. (40,50,60,70,80)<br>b. (40,50,60,70)<br>c. [40,60]<br>d. (40,60)                                                                                                                                                                                      |
| 4    | Which of the following option is not correct?<br>a. if we try to read a text file that does not exist, an error occurs.<br>b. if we try to read a text file that does not exist, the file gets created.<br>c. if we try to write on a text file that does not exist, no error occurs.<br>d. if we try to write on a text file that does not exist, the file gets Created. |
| 5    | Which of the following options can be used to read the first line of a text file Myfile.txt?<br>a. myfile = open('Myfile.txt'); myfile.read()<br>b. myfile = open('Myfile.txt','r'); myfile.read(n)<br>c. myfile = open('Myfile.txt'); myfile.readline()<br>d. myfile = open('Myfile.txt'); myfile.readlines()                                                            |

|    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 6  | <p>Assume that the position of the file pointer is at the beginning of 3rd line in a text file. Which of the following option can be used to read all the remaining lines?</p> <ul style="list-style-type: none"> <li>a. myfile.read()</li> <li>b. myfile.read(n)</li> <li>c. myfile.readline()</li> <li>d. myfile.readlines()</li> </ul>                                                                                                                                                                                          |
| 7  | <p>A text file student.txt is stored in the storage device. Identify the correct option out of the following options to open the file in read mode.</p> <ul style="list-style-type: none"> <li>i. myfile = open('student.txt','rb')</li> <li>ii. myfile = open('student.txt','w')</li> <li>iii. myfile = open('student.txt','r')</li> <li>iv. myfile = open('student.txt')</li> </ul> <ul style="list-style-type: none"> <li>a. only i</li> <li>b. both i and iv</li> <li>c. both iii and iv</li> <li>d. both i and iii</li> </ul> |
| 8  | <p>The return type of the input() function is</p> <ul style="list-style-type: none"> <li>a. string</li> <li>b. integer</li> <li>c. list</li> <li>d. tuple</li> </ul>                                                                                                                                                                                                                                                                                                                                                               |
| 9  | <p>Which of the following operator cannot be used with string data type?</p> <ul style="list-style-type: none"> <li>a. +</li> <li>b. in</li> <li>c. *</li> <li>d. /</li> </ul>                                                                                                                                                                                                                                                                                                                                                     |
| 10 | <p>Consider a tuple tup1 = (10, 15, 25, and 30). Identify the statement that will result in an error.</p> <ul style="list-style-type: none"> <li>a. print(tup1[2])</li> <li>b. tup1[2] = 20</li> <li>c. print(min(tup1))</li> <li>d. print(len(tup1))</li> </ul>                                                                                                                                                                                                                                                                   |
| 11 | <p>Which of the following statement is incorrect in the context of binary files?</p> <ul style="list-style-type: none"> <li>a. Information is stored in the same format in which the information is held in memory.</li> <li>b. No character translation takes place</li> <li>c. Every line ends with a new line character</li> <li>d. pickle module is used for reading and writing</li> </ul>                                                                                                                                    |
| 12 | <p>What is the significance of the tell() method?</p> <ul style="list-style-type: none"> <li>a. tells the path of file</li> <li>b. tells the current position of the file pointer within the file</li> <li>c. tells the end position within the file</li> <li>d. checks the existence of a file at the desired location</li> </ul>                                                                                                                                                                                                 |

|    |                                                                                                                                                                                                                                                                                                                                                                                                                            |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 13 | <p>Which of the following statement is true?</p> <ul style="list-style-type: none"> <li>a. pickling creates an object from a sequence of bytes</li> <li>b. pickling is used for object serialization</li> <li>c. pickling is used for object deserialization</li> <li>d. pickling is used to manage all types of files in Python</li> </ul>                                                                                |
| 14 | <p>Syntax of seek function in Python is <code>myfile.seek(offset, reference_point)</code> where <code>myfile</code> is the file object. What is the default value of <code>reference_point</code>?</p> <ul style="list-style-type: none"> <li>a. 0</li> <li>b. 1</li> <li>c. 2</li> <li>d. 3</li> </ul>                                                                                                                    |
| 15 | <p>Which of the following components are part of a function header in Python?</p> <ul style="list-style-type: none"> <li>a. Function Name</li> <li>b. Return Statement</li> <li>c. Parameter List</li> <li>d. Both a and c</li> </ul>                                                                                                                                                                                      |
| 16 | <p>Which of the following function header is correct?</p> <ul style="list-style-type: none"> <li>a. <code>def cal_si(p=100, r, t=2)</code></li> <li>b. <code>def cal_si(p=100, r=8, t)</code></li> <li>c. <code>def cal_si(p, r=8, t)</code></li> <li>d. <code>def cal_si(p, r=8, t=2)</code></li> </ul>                                                                                                                   |
| 17 | <p>Which of the following is the correct way to call a function?</p> <ul style="list-style-type: none"> <li>a. <code>my_func()</code></li> <li>b. <code>def my_func()</code></li> <li>c. <code>return my_func</code></li> <li>d. <code>call my_func()</code></li> </ul>                                                                                                                                                    |
| 18 | <p>Which of the following character acts as default delimiter in a csv file?</p> <ul style="list-style-type: none"> <li>a. (colon) :</li> <li>b. (hyphen) -</li> <li>c. (comma) ,</li> <li>d. (vertical line)  </li> </ul>                                                                                                                                                                                                 |
| 19 | <p>Syntax for opening Student.csv file in write mode is <code>myfile = open("Student.csv","w",newline=)</code>.</p> <p>What is the importance of <code>newline=</code>?</p> <ul style="list-style-type: none"> <li>a. A newline gets added to the file</li> <li>b. Empty string gets appended to the first line.</li> <li>c. Empty string gets appended to all lines.</li> <li>d. EOL translation is suppressed</li> </ul> |
| 20 | <p>What is the correct expansion of CSV files?</p> <ul style="list-style-type: none"> <li>a. Comma Separable Values</li> <li>b. Comma Separated Values</li> <li>c. Comma Split Values</li> <li>d. Comma Separation Values</li> </ul>                                                                                                                                                                                       |

|    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 21 | <p>Which of the following is not a function / method of csv module in Python?</p> <ul style="list-style-type: none"> <li>a. read()</li> <li>b. reader()</li> <li>c. writer()</li> <li>d. writerow()</li> </ul>                                                                                                                                                                                                                                                                                                                      |
| 22 | <p>Which one of the following is the default extension of a Python file?</p> <ul style="list-style-type: none"> <li>a. .exe</li> <li>b. .p++</li> <li>c. .py</li> <li>d. .p</li> </ul>                                                                                                                                                                                                                                                                                                                                              |
| 23 | <p>Which of the following symbol is used in Python for single line comment?</p> <ul style="list-style-type: none"> <li>a. /</li> <li>b. /*</li> <li>c. //</li> <li>d. #</li> </ul>                                                                                                                                                                                                                                                                                                                                                  |
| 24 | <p>Which of the following statement opens a binary file record.bin in write mode and writes data from a list lst1 = [1,2,3,4] on the binary file?</p> <ul style="list-style-type: none"> <li>a. with open('record.bin','wb') as myfile:<br/>pickle.dump(lst1,myfile)</li> <li>b. with open('record.bin','wb') as myfile:<br/>pickle.dump(myfile,lst1)</li> <li>c. with open('record.bin','wb+') as myfile:<br/>pickle.dump(myfile,lst1)</li> <li>d. with open('record.bin','ab') as myfile:<br/>pickle.dump(myfile,lst1)</li> </ul> |
| 25 | <p>Which of these about a dictionary is false?</p> <ul style="list-style-type: none"> <li>a) The values of a dictionary can be accessed using keys</li> <li>b) The keys of a dictionary can be accessed using values</li> <li>c) Dictionaries aren't ordered</li> <li>d) Dictionaries are mutable</li> </ul>                                                                                                                                                                                                                        |
|    | <b>Section-B</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|    | <b>This section consists of 24 Questions (26 to 49). Attempt any 20 questions.</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 26 | <p>What is the output of following code:</p> <pre>T=(100) print(T*2)</pre> <ul style="list-style-type: none"> <li>a. Syntax error</li> <li>b. (200,)</li> <li>c. 200</li> <li>d. (100,100)</li> </ul>                                                                                                                                                                                                                                                                                                                               |

|    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 27 | <p>Suppose content of 'Myfile.txt' is:</p> <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> Twinkle twinkle little star<br/> How I wonder what you are<br/> Up above the world so high<br/> Like a diamond in the sky </div> <p>What will be the output of the following code?</p> <pre>myfile = open("Myfile.txt") data = myfile.readlines() print(len(data)) myfile.close()</pre> <p>a. 3<br/>b. 4<br/>c. 5<br/>d. 6</p> |
| 28 | <p>Identify the output of the following Python statements.</p> <pre>x = [[10.0, 11.0, 12.0], [13.0, 14.0, 15.0]] y = x[1][2] print(y)</pre> <p>a. 12.0<br/>b. 13.0<br/>c. 14.0<br/>d. 15.0</p>                                                                                                                                                                                                                                                                        |
| 29 | <p>Identify the output of the following Python statements.</p> <pre>x = 2 while x &lt; 9:     print(x, end='')     x = x + 1</pre> <p>a. 12345678<br/>b. 123456789<br/>c. 2345678<br/>d. 23456789</p>                                                                                                                                                                                                                                                                 |
| 30 | <p>Identify the output of the following Python statements.</p> <pre>b = 1 for a in range(1, 10, 2):     b += a + 2 print(b)</pre> <p>a. 31<br/>b. 33<br/>c. 36<br/>d. 39</p>                                                                                                                                                                                                                                                                                          |

| 31    | <p>Identify the output of the following Python statements.</p> <pre>lst1 = [10, 15, 20, 25, 30] lst1.insert( 3, 4) lst1.insert( 2, 3) print (lst1[-5])</pre> <p>a. 2<br/>b. 3<br/>c. 4<br/>d. 20</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |        |         |        |     |      |       |     |         |       |     |        |       |     |       |       |     |              |       |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|---------|--------|-----|------|-------|-----|---------|-------|-----|--------|-------|-----|-------|-------|-----|--------------|-------|
| 32    | <p>Raghav is trying to write a tuple tup1 = (1,2,3,4,5) on a binary file <b>test.bin</b>. Consider the following code written by him.</p> <pre>import pickle tup1 = (1,2,3,4,5) myfile = open("test.bin", 'wb') pickle._____ #Statement 1 myfile.close()</pre> <p>Identify the missing code in Statement 1.</p> <p>a. dump(myfile,tup1)<br/>b. dump(tup1, myfile)<br/>c. write(tup1,myfile)<br/>d. load(myfile,tup1)</p>                                                                                                                                                                                                                                                                                                                                                                                                                                            |        |         |        |     |      |       |     |         |       |     |        |       |     |       |       |     |              |       |
| 33    | <p>A binary file employee.dat has following data</p> <table><tr><th>Empno</th><th>empname</th><th>Salary</th></tr><tr><td>101</td><td>Anuj</td><td>50000</td></tr><tr><td>102</td><td>Arijita</td><td>40000</td></tr><tr><td>103</td><td>Hanika</td><td>30000</td></tr><tr><td>104</td><td>Firoz</td><td>60000</td></tr><tr><td>105</td><td>Vijaylakshmi</td><td>40000</td></tr></table> <pre>def display(eno):     f=open("employee.dat", "rb")     totSum=0     try:         while True:             R=pickle.load(f)             if R[0]==eno:                 _____ #Line1                 totSum=totSum+R[2]     except:         f.close()     print(totSum)</pre> <p>When the above mentioned function, display (103) is executed, the output displayed is 190000.</p> <p>Write appropriate jump statement from the following to obtain the above output.</p> | Empno  | empname | Salary | 101 | Anuj | 50000 | 102 | Arijita | 40000 | 103 | Hanika | 30000 | 104 | Firoz | 60000 | 105 | Vijaylakshmi | 40000 |
| Empno | empname                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Salary |         |        |     |      |       |     |         |       |     |        |       |     |       |       |     |              |       |
| 101   | Anuj                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 50000  |         |        |     |      |       |     |         |       |     |        |       |     |       |       |     |              |       |
| 102   | Arijita                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 40000  |         |        |     |      |       |     |         |       |     |        |       |     |       |       |     |              |       |
| 103   | Hanika                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 30000  |         |        |     |      |       |     |         |       |     |        |       |     |       |       |     |              |       |
| 104   | Firoz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 60000  |         |        |     |      |       |     |         |       |     |        |       |     |       |       |     |              |       |
| 105   | Vijaylakshmi                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 40000  |         |        |     |      |       |     |         |       |     |        |       |     |       |       |     |              |       |



|    |                                                                                                                                                                                                                                                                                                                                                                         |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <ul style="list-style-type: none"> <li>a. jump</li> <li>b. break</li> <li>c. continue</li> <li>d. return</li> </ul>                                                                                                                                                                                                                                                     |
| 34 | <p>What will be the output of the following Python code?</p> <pre>def add (num1, num2):     sum = num1 + num2 sum = add(20,30) print(sum)</pre> <ul style="list-style-type: none"> <li>a. 50</li> <li>b. 0</li> <li>c. Null</li> <li>d. None</li> </ul>                                                                                                                 |
| 35 | <p>Evaluate the following expression and identify the correct answer.</p> <p><math>16 - (4 + 2) * 5 + 2 ** 3 * 4</math></p> <ul style="list-style-type: none"> <li>a. 54</li> <li>b. 46</li> <li>c. 18</li> <li>d. 32</li> </ul>                                                                                                                                        |
| 36 | <p>What will be the output of the following code?</p> <pre>def my_func(var1=100, var2=200):     var1+=10     var2 = var2 - 10     return var1+var2 print(my_func(50),my_func())</pre> <ul style="list-style-type: none"> <li>a. 100 200</li> <li>b. 150 300</li> <li>c. 250 75</li> <li>d. 250 300</li> </ul>                                                           |
| 37 | <p>What will be the output of the following code?</p> <pre>value = 50 def display(N):     global value     value = 25     if N%7==0:         value = value + N     else:         value = value - N print(value, end="#") display(20) print(value)</pre> <ul style="list-style-type: none"> <li>a. 50#50</li> <li>b. 50#5</li> <li>c. 50#30</li> <li>d. 5#50#</li> </ul> |

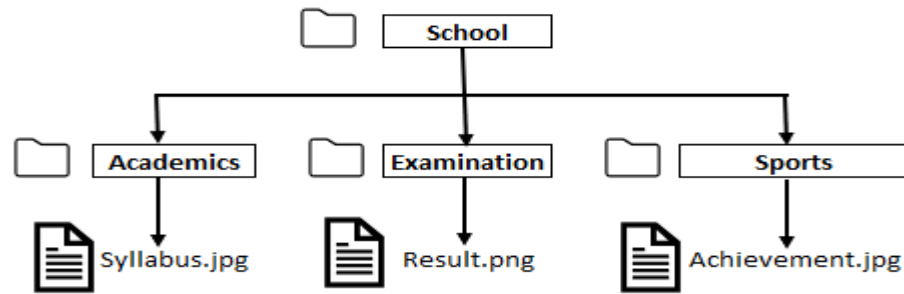
|    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 38 | <p>What will be the output of the following code?</p> <pre>import random List=["Delhi","Mumbai","Chennai","Kolkata"] for y in range(4):     x = random.randint(1,3)     print(List[x],end="#")</pre> <p>a. Delhi#Mumbai#Chennai#Kolkata#<br/> b. Mumbai#Chennai#Kolkata#Mumbai#<br/> c. Mumbai# Mumbai #Mumbai # Delhi#<br/> d. Mumbai# Mumbai #Chennai # Mumbai</p>                                                                                                                        |
| 39 | <p>What is the output of the following code snippet?</p> <pre>def ChangeVal(M,N):     for i in range(N):         if M[i]%5 == 0:             M[i]//=5         if M[i]%3 == 0:             M[i]//=3 L = [25,8,75,12] ChangeVal(L,4) for i in L:     print(i,end="#")</pre> <p>a) 5#8#15#4#<br/> b) 5#8#5#4#<br/> c) 5#8#15#14#<br/> d) 5#18#15#4#</p>                                                                                                                                        |
| 40 | <p>Suppose content of 'Myfile.txt' is</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> Humpty Dumpty sat on a wall<br/> Humpty Dumpty had a great fall<br/> All the king's horses and all the king's men<br/> Couldn't put Humpty together again </div> <p>What will be the output of the following code?</p> <pre>myfile = open("Myfile.txt") record = myfile.read().split() print(len(record)) myfile.close()</pre> <p>a. 24<br/> b. 25<br/> c. 26<br/> d. 27</p> |
| 41 | <p>Find the output of the following code:</p> <pre>Name="Python3.1" R="" for x in range(len(Name)):     if Name[x].isupper():         R=R+Name[x].lower()</pre>                                                                                                                                                                                                                                                                                                                             |

|    |                                                                                                                                                                                                                                                                                                                                                                                           |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <pre> elif Name[x].islower():     R=R+Name[x].upper() elif Name[x].isdigit():     R=R+Name[x-1] else:     R=R+"#" print(R) </pre> <p> a. pYTHOn##@<br/> b. pYTHOnN#@<br/> c. pYTHOn#@<br/> d. pYTHOnN@# </p>                                                                                                                                                                              |
| 42 | <p>Suppose content of 'Myfile.txt' is</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;">Honesty is the best policy.</div> <p>What will be the output of the following code?</p> <pre> myfile = open("Myfile.txt") x = myfile.read() print(len(x)) myfile.close() </pre> <p> a. 5<br/> b. 25<br/> c. 26<br/> d. 27 </p>                                       |
| 43 | <p>Suppose content of 'Myfile.txt' is</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;">Culture is the widening of the mind and of the spirit.</div> <p>What will be the output of the following code?</p> <pre> myfile = open("Myfile.txt") x = myfile.read() y = x.count('the') print(y) myfile.close() </pre> <p> a. 2<br/> b. 3<br/> c. 4<br/> d. 5 </p> |
| 44 | <p>What will be the output of the following code?</p> <pre> x = 3 def myfunc():     global x     x+=2     print(x, end=' ') print(x, end=' ') myfunc() print(x, end=' ') </pre>                                                                                                                                                                                                           |

|    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <p>a. 3 3 3<br/>b. 3 4 5<br/>c. 3 3 5<br/>d. 3 5 5</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 45 | <p>Suppose content of 'Myfile.txt' is</p> <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <p>Ek Bharat Shreshtha Bharat</p> </div> <p>What will be the output of the following code?</p> <pre> myfile = open("Myfile.txt") vlist = list("aeiouAEIOU") vc=0 x = myfile.read() for y in x:     if(y in vlist):         vc+=1 print(vc) myfile.close() </pre> <p>a. 6<br/>b. 7<br/>c. 8<br/>d. 9</p>                                                                                                                                               |
| 46 | <p>Suppose content of 'Myfile.txt' is</p> <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <p>Twinkle twinkle little star<br/>How I wonder what you are<br/>Up above the world so high<br/>Like a diamond in the sky<br/>Twinkle twinkle little star</p> </div> <p>What will be the output of the following code?</p> <pre> myfile = open("Myfile.txt") line_count = 0 data = myfile.readlines() for line in data:     if line[0] == 'T':         line_count += 1 print(line_count) myfile.close() </pre> <p>a. 2<br/>b. 3<br/>c. 4<br/>d. 5</p> |

47

Consider the following directory structure.



Suppose root directory (School) and present working directory are the same. What will be the absolute path of the file Syllabus.jpg?

- a. School/syllabus.jpg
- b. School/Academics/syllabus.jpg
- c. School/Academics/../syllabus.jpg
- d. School/Examination/syllabus.jpg

48

Assume the content of text file, 'student.txt' is:

```
Arjun Kumar
Ismail Khan
Joseph B
Hanika Kiran
```

What will be the data type of data\_rec?

```
myfile = open("Myfile.txt")
data_rec = myfile.readlines()
myfile.close()
```

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

49

What will be the output of the following code?

```
tup1 = (1,2,[1,2],3)
tup1[2][1]=3.14
print(tup1)
```

- a. (1,2,[3.14,2],3)
- b. (1,2,[1,3.14],3)
- c. (1,2,[1,2],3.14)
- d. Error Message

|    | <p align="center"><b>Section-C</b><br/><b>Case Study based Questions</b></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <p align="center"><b>This section consists of 6 Questions (50 -55) Attempt any 5 questions.</b></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|    | <p>Rohit, a student of class 12, is learning CSV File Module in Python. During examination, he has been assigned an incomplete python code (shown below) to create a CSV File 'Student.csv' (content shown below). Help him in completing the code which creates the desired CSV File.</p> <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <p><b>CSV File</b><br/> 1,AKSHAY,XII,A<br/> 2,ABHISHEK,XII,A<br/> 3,ARVIND,XII,A<br/> 4,RAVI,XII,A<br/> 5,ASHISH,XII,A</p> </div> <p><b>Incomplete Code</b></p> <pre> import _____ #Statement-1 fh = open(_____, _____, newline="") #Statement-2 stuwriter = csv._____ #Statement-3 data = [ ] header = ['ROLL_NO', 'NAME', 'CLASS', 'SECTION'] data.append(header) for i in range(5):     roll_no = int(input("Enter Roll Number : "))     name = input("Enter Name : ")     Class = input("Enter Class : ")     section = input("Enter Section : ")     rec = [ _____ ] #Statement-4     data.append(_____) #Statement-5 stuwriter. _____ (data) #Statement-6 fh.close() </pre> |
| 50 | <p>Identify the suitable code for blank space in the line marked as Statement-1.</p> <p>a) csv file<br/> b) CSV<br/> c) csv<br/> d) cvs</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 51 | <p>Identify the missing code for blank space in line marked as Statement-2.</p> <p>a) "Student.csv","wb"<br/> b) "Student.csv","w"<br/> c) "Student.csv","r"<br/> d) "Student.cvs","r"</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 52 | <p>Choose the function name (with argument) that should be used in the blank space of line marked as Statement-3.</p> <p>a) reader(fh)<br/> b) reader(MyFile)<br/> c) writer(fh)<br/> d) writer(MyFile)</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

|    |                                                                                                                                                                                                                                                                                                             |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    |                                                                                                                                                                                                                                                                                                             |
| 53 | <p>Identify the suitable code for blank space in line marked as Statement-4.</p> <ul style="list-style-type: none"> <li>a) 'ROLL_NO', 'NAME', 'CLASS', 'SECTION'</li> <li>b) ROLL_NO, NAME, CLASS, SECTION</li> <li>c) 'roll_no','name','Class','section'</li> <li>d) roll_no,name,Class,section</li> </ul> |
| 54 | <p>Identify the suitable code for blank space in the line marked as Statement-5.</p> <ul style="list-style-type: none"> <li>a) data</li> <li>b) record</li> <li>c) rec</li> <li>d) insert</li> </ul>                                                                                                        |
| 55 | <p>Choose the function name that should be used in the blank space of line marked as Statement-6 to create the desired CSV File?</p> <ul style="list-style-type: none"> <li>a) dump()</li> <li>b) load()</li> <li>c) writerows()</li> <li>d) writerow()</li> </ul>                                          |

**Marking Scheme**  
**Class: XII Session: 2021-22**  
**Computer Science (Code 083)**

|    |                                                                              |
|----|------------------------------------------------------------------------------|
| 1  | d. pass                                                                      |
| 2  | b. tuple                                                                     |
| 3  | d. (40,60)                                                                   |
| 4  | b. if we try to read a text file that does not exist, the file gets created. |
| 5  | c. myfile = open('Myfile.txt'); myfile.readline()                            |
| 6  | d. myfile.readlines()                                                        |
| 7  | c. both iii and iv                                                           |
| 8  | a. string                                                                    |
| 9  | d. /                                                                         |
| 10 | b. tup1[2] = 20                                                              |
| 11 | c. Every line ends with a new line character                                 |
| 12 | b. tells the current position of the file pointer within the file            |
| 13 | b. pickling is used for object serialization                                 |
| 14 | a. 0                                                                         |
| 15 | d. Both a and c                                                              |
| 16 | d. def cal_si(p, r=8, t=2)                                                   |
| 17 | a. my_func()                                                                 |
| 18 | c. ,                                                                         |
| 19 | d. EOL Translation is suppressed                                             |
| 20 | b. Comma Separated Values                                                    |
| 21 | a. read()                                                                    |
| 22 | c. .py                                                                       |
| 23 | d. #                                                                         |
| 24 | a. with open('record.bin','wb') as myfile:<br>pickle.dump(lst1,myfile)       |
| 25 | b. The keys of a dictionary can be accessed using values                     |
| 26 | c. 200                                                                       |
| 27 | b. 4                                                                         |
| 28 | d. 15.0                                                                      |
| 29 | c. 2345678                                                                   |
| 30 | c. 36                                                                        |
| 31 | b. 3                                                                         |
| 32 | b. dump(tup1, myfile)                                                        |
| 33 | c. continue                                                                  |
| 34 | d. None                                                                      |
| 35 | c. 18                                                                        |
| 36 | d. 250 300                                                                   |
| 37 | b. 50#5                                                                      |
| 38 | b. Mumbai#Chennai#Kolkata#Mumbai#                                            |
| 39 | b. 5#8#5#4#                                                                  |
| 40 | c. 26                                                                        |
| 41 | b. pYTHOnN#@                                                                 |
| 42 | d. 27                                                                        |



|    |                                  |
|----|----------------------------------|
| 43 | b. 3                             |
| 44 | d. 3 5 5                         |
| 45 | b. 7                             |
| 46 | a. 2                             |
| 47 | b. School/Academics/syllabus.jpg |
| 48 | b. list                          |
| 49 | b. (1,2,[1,3.14],3)              |
| 50 | c. csv                           |
| 51 | b. "Student.csv", "w"            |
| 52 | c. writer(fh)                    |
| 53 | d. roll_no,name,Class,section    |
| 54 | c. rec                           |
| 55 | c. writerows()                   |

**KENDRIYA VIDYALAYA SANGATHAN, ERNAKULAM REGION**  
**TERM I-PRACTICE TEST 2021-22**

**Class: XII**

**Computer Science – NEW- PYTHON (083)**

**(Theory: Term-1 MARKING SCHEME)**

**Maximum Marks: 35**

**Time Allowed: 90 minutes**

**General Instructions:**

- ☐ The question paper is divided into 3 Sections - A, B and C.
- ☐ Section A, consist of 25 Questions (1-25). Attempt any 20 questions.
- ☐ Section B, consist of 24 Questions (26-49). Attempt any 20 questions.
- ☐ Section C, consist of 6 case study based Questions (50-55). Attempt any 5 questions.
- ☐ **All questions carry equal marks(0.77)**

| Q No | SECTION A                                                                                                                                                                                                                                                                                                                                              |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|      | <b>This section consists of 25 Questions (1 to 25). Attempt any 20 questions from this section. Choose the best possible option.</b>                                                                                                                                                                                                                   |
| 1    | Identify the invalid identifier given below.<br><br>a) _2017discount                      b) Profit                      c) Total-discount                      d) Totaldiscount<br><br><b>Ans :c</b>                                                                                                                                                  |
| 2    | Multiline comments can be added using _____ on each end of the comment.<br><br>a) "" "" (triple quote)                      b) # (Hash)                      c) \$ (dollar)                      d) % (modulus)<br><b>Ans: a</b>                                                                                                                       |
| 3    | The operators is and is not are _____.<br><br>a) Identity Operators                      b) Comparison Operators<br>c) Membership Operators                      d) Unary Operators<br><b>Ans : a</b>                                                                                                                                                  |
| 4    | Which of the following is/are valid declaration of a dictionary?<br><br>a) D = {'StuName': 'Alan', 'StuAge': 30, 'StuCity': 'Vizag'}<br>b) D = ['StuName': 'Alan', 'StuAge': 30, 'StuCity': 'Vizag']<br>c) D = ('StuName': 'Alan', 'StuAge': 30, 'StuCity': 'Vizag')<br>d) D = {'StuName'; 'Alan', 'StuAge': 30, 'StuCity': 'Vizag'}<br><b>Ans : a</b> |
| 5    | Given the lists L=["K", "V", "S", "E", "R", "K"], write the output of print(L[1:4])                                                                                                                                                                                                                                                                    |

|    |                                                                                                                                                                                                                                                                                                                                                                                                            |
|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <p>a) ['K', 'V', 'S']                      b) ['V', 'S', 'E']<br/> c) ['K', 'V', 'S', 'E']                  d) ['V', 'S', 'E', 'R']</p> <p><b>Ans : b</b></p>                                                                                                                                                                                                                                              |
| 6  | <p>A local variable in Python is a variable that is,</p> <p>a. Defined inside every function                      b. Local to the given program<br/> c. Accessible from within the function                  d. All of these</p> <p><b>Ans : d</b></p>                                                                                                                                                     |
| 7  | <p>Which keyword is used to define a function?</p> <p>a)Fun                      b)Define                      c)def                      d)function</p> <p><b>Ans : c</b></p>                                                                                                                                                                                                                             |
| 8  | <p>To import specific functions in a program from a module, ..... statement can be used.</p> <p>a)with                      b)from                      c)import                      d)open</p> <p><b>Ans : b</b></p>                                                                                                                                                                                     |
| 9  | <p>Multiple values from a function in Python are returned through the following:</p> <p>a)List                      b)String                      c)Tuple                      d)All of these</p> <p><b>Ans : c</b></p>                                                                                                                                                                                    |
| 10 | <p>If return statement is not used inside the function, the function will return:</p> <p>a) None                      b) 0                      c) Null                      d) Arbitrary value</p> <p><b>Ans : a</b></p>                                                                                                                                                                                  |
| 11 | <p>Which of the following commands can be used to read “n” number of characters from a file using the file object &lt;file&gt;?</p> <p>a) file.read(n)      b) n = file.read()      c). file.readline(n)      d) file.readlines()</p> <p><b>Ans : a</b></p>                                                                                                                                                |
| 12 | <p>CSV module allows to write multiple rows using _____ function.</p> <p>a. writerows( )   b. writerow( )   c. writer( )   d. None of the above</p> <p><b>Ans : a</b></p>                                                                                                                                                                                                                                  |
| 13 | <p>By default a file in python is opened in .....mode.</p> <p>a)read                      b)write                      c)read and write                      d)binary</p> <p><b>Ans : a</b></p>                                                                                                                                                                                                            |
| 14 | <p>What does the &lt;readlines()&gt; method returns?</p> <p>a) string                      b) a list of strings                      c) list of single characters                      d) list of integers</p> <p><b>Ans : b</b></p>                                                                                                                                                                       |
| 15 | <p>Which of the following statements are true?</p> <p>a) When you open a file for reading, if the file does not exist, an error occurs<br/> b) When you open a file for writing, if the file does not exist, a new file is created<br/> c) When you open a file for writing, if the file exists, the existing file is overwritten with the new file<br/> d) All of the mentioned</p> <p><b>Ans : d</b></p> |

|    |                                                                                                                                                                                                                                                                                                            |
|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 16 | <p>Choose the correct output of the following Python Statement:</p> <pre>print(math.ceil(81.3))</pre> <p>a)80                      b)81                      c)82                      d)81.3</p> <p><b>Ans : c</b></p>                                                                                    |
| 17 | <p>The file paths from the topmost level of the directory structure is known as</p> <p>a)File path              b)Absolute path      c)Relative Path      d)None</p> <p><b>Ans : b</b></p>                                                                                                                 |
| 18 | <p>Consider a file named rome.txt, then the statement used to open a file for reading,</p> <p>a. infile = open("c:\rome.txt", "r")<br/> b. infile = open("c:\\rome.txt", "r")<br/> c. infile = open(file = "c:\rome.txt", "r")<br/> d. infile = open(file = "c:\\rome.txt", "r")</p> <p><b>Ans : b</b></p> |
| 19 | <p>Which of the following results in a Syntax Error ?</p> <p>a) ‘‘Once upon a time...’, she said.’                      b) ‘‘He said, ‘Yes!’’’<br/> c) ‘3\’                      d) ‘’’That’s okay’’’</p> <p><b>Ans : c</b></p>                                                                            |
| 20 | <p>Evaluate the expression and choose the correct answer: <math>9-5//2+7**2-1</math></p> <p>a)53                      b)51                      c)55                      d)57</p> <p><b>Ans : c</b></p>                                                                                                   |
| 21 | <p>What will be the output of the following Python expression?</p> <pre>print(round(4.576))</pre> <p>a)4                      b)4.6                      c)5                      d)4.5</p> <p><b>Ans : c</b></p>                                                                                          |
| 22 | <p>Consider the following function headers. Identify the correct statement: -</p> <p>a) def correct(a=1,b=2,c):                      b) def correct(a=1,b,c=3):<br/> c) def correct(a=1,b=2,c=3):                      d) def correct(a=1,b,c):</p> <p><b>Ans : c</b></p>                                  |
| 23 | <p>Which one of the following is the default extension of a Python file?</p> <p>a. .exe<br/> b. .p+<br/> c. .py<br/> d. .python</p> <p><b>Ans : c</b></p>                                                                                                                                                  |
| 24 | <p>Which of the following character acts as default delimiter in a csv file?</p> <p>a. (colon) :</p>                                                                                                                                                                                                       |



|                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                  |                   |                  |                   |         |  |  |  |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|-------------------|------------------|-------------------|---------|--|--|--|
|                 | <div>a,b=5,10<br/>x=20<br/>fun1(b)<br/>print(x)</div> <table><tr><td>a)16<br/>20<br/>4</td><td>b) 5<br/>16<br/>4</td><td>c) 20<br/>16<br/>4</td><td>d) 20<br/>16<br/>20</td></tr><tr><td>Ans : b</td><td></td><td></td><td></td></tr></table>                                                                                                                                                                                                                        | a)16<br>20<br>4  | b) 5<br>16<br>4   | c) 20<br>16<br>4 | d) 20<br>16<br>20 | Ans : b |  |  |  |
| a)16<br>20<br>4 | b) 5<br>16<br>4                                                                                                                                                                                                                                                                                                                                                                                                                                                      | c) 20<br>16<br>4 | d) 20<br>16<br>20 |                  |                   |         |  |  |  |
| Ans : b         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                  |                   |                  |                   |         |  |  |  |
| 29              | <div>What will be the output generated by the following snippet?<br/><br/>a = [5, 10, 15, 20, 25]<br/>k = 1<br/>i = a[1] + 1<br/>j = a[2] +1<br/>m = a[k+1]<br/>print (i, j, m)<br/><br/>a) 11 15 16                      b) 11 16 15<br/>c) 11 15 15                      d) 16 11 15<br/><br/>Ans : b</div>                                                                                                                                                        |                  |                   |                  |                   |         |  |  |  |
| 30              | <div>What will be the output of following code:<br/><br/>def Alter(x, y = 10, z=20):<br/>    sum1=x+y+z<br/>    print(sum1)<br/>Alter(10,20,30)<br/>Alter(20,30)<br/>Alter(100)<br/><br/>a)50                      b)60                      c)60                      d)50<br/>60                      70                      50                      70<br/>120                      130                      120                      120<br/><br/>Ans : b</div> |                  |                   |                  |                   |         |  |  |  |
| 31              | <div>Find the correct code after removing the errors: -<br/>    def SI(p,t=2,r):<br/>        return (p*r*t)/100<br/><br/>a)def SI(p, ,r):<br/>    return(p*r*t)/100<br/><br/>b) Def SI(p,t,r):<br/>    return(p*r*t)/100<br/><br/>c)def SI(p,r,t=2):</div>                                                                                                                                                                                                           |                  |                   |                  |                   |         |  |  |  |

|    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <pre>return(p*r*t)/100</pre> <p>d) def SI(t=2,p,r):</p> <pre>return(p*r*t)/100</pre> <p><b>Ans : c</b></p>                                                                                                                                                                                                                                                                                                                                                                    |
| 32 | <p>Given the S = "KENDRIYA VIDYALAYA ", write the output of print(S[-1:-10:-2])</p> <p>a)KNRY                      b) AYALA                      c) AAADV                      d) YLYI</p> <p><b>Ans : c</b></p>                                                                                                                                                                                                                                                              |
| 33 | <p>Complete the given python code to create and add the given content to a binary file.</p> <pre>import pickle listvalues=[1,"Geetika",'F', 26] fileobject=open("mybinary.dat", "wb") pickle . .....(listvalues,.....)                      #line1 fileobject.close()</pre> <p>a)load ,fileobject    b)open, fileobject    c)dump, fileobject    d)fileobject, dump</p> <p><b>Ans : c</b></p>                                                                                 |
| 34 | <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> <p>a) SCHOOLbbbbCOM                      b) SCHOOL@ @ @COM</p> <p>c) Shool2bbbbcom                      d) Shoolbbbbcom</p> <p><b>Ans : c</b></p> |
| 35 | <p>Suppose content of 'Myfile.txt' is</p> <div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <p><b>We may be young or old,<br/>But we must be bold,<br/>Against the enemy untold.<br/>If you have to go out, think twice,<br/>Wear your mask, be wise.<br/>After coming home, sanitise,<br/>And wash your hands thrice.</b></p> </div>                                                                                              |

|    |                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <p>What will be the output of the following code?</p> <pre> myfile = open("Myfile.txt") line_count = 0 data = myfile.readlines() for line in data:     if line[0] == 'W':         line_count += 1 print(line_count) myfile.close() </pre> <p>a) 1      b) 2      c) 3      d) 4</p> <p><b>Ans : b</b></p>                                                                                                                                 |
| 36 | <p>What will be the output of the following code?</p> <pre> tup1 = (1,2,[1,2],3) tup1[2][1]=3.14 print(tup1) </pre> <p>a. (1,2,[3.14,2],3)<br/> b. (1,2,[1,3.14],3)<br/> c. (1,2,[1,2],3.14)<br/> d. Error Message</p> <p><b>Ans : b</b></p>                                                                                                                                                                                              |
| 37 | <p>Hari is trying to write a tuple tup1 = (1,2,3,4,5) on a binary file test.bin. Consider the following code written by him.</p> <pre> import pickle tup1 = (1,2,3,4,5) myfile = open("test.bin",'wb') pickle._____ #Statement 1 myfile.close() </pre> <p>Identify the missing code in Statement 1.</p> <p>a. dump(myfile,tup1)<br/> b. dump(tup1, myfile)<br/> c. write(tup1,myfile)<br/> d. load(myfile,tup1)</p> <p><b>Ans : b</b></p> |
| 38 | <p>What will be the output of the following code</p> <pre> str1="I love Python" strlen=len(str1)+5 print(strlen) </pre> <p>a)18      b) 19      c) 13      d) 15</p>                                                                                                                                                                                                                                                                      |



|    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <b>Ans : a</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 39 | <p>What is the output of the program given below:</p> <pre>x=50 def func(x):     x=2 func(x) print('x is now',x)</pre> <p>a) x is now 50      b) x is now 2<br/>b) x is now 100    d) Error</p> <p><b>Ans : a</b></p>                                                                                                                                                                                                                                                                            |
| 40 | <p>What is the output of the following statement ?</p> <pre>print("xyyzxyzxxyy".count('yy',1))</pre> <p>a)2      b)0      c)1      d)error</p> <p><b>Ans : b</b></p>                                                                                                                                                                                                                                                                                                                             |
| 41 | <p>Assertion - close () method closes the file and frees the memory space acquired by that file.<br/>Reason - one should always close their files, in some cases, due to buffering, changes made to a file may not show until the file is closed.</p> <p>A - Both A and R are true and R is the correct explanation of A.<br/>B - Both A and R are true but R is not the correct explanation of A.<br/>C - A is true but R is false.<br/>D - A is false but R is true.</p> <p><b>Ans : b</b></p> |
| 41 | <p>What is the output of below program?</p> <pre>def say(message, times = 1):     print(message * times) say('Hello') say('World', 5)</pre> <p>a) Hello<br/>WorldWorldWorldWorldWorld<br/>b) Hello World 5<br/>c) Hello World,World,World,World,World<br/>d) Hello HelloHelloHelloHelloHello</p> <p><b>Ans : a</b></p>                                                                                                                                                                           |
| 43 | <p>Which statement will read 5 characters from a file(file object 'f')?</p> <p>a)f.read()<br/>b) f.read(5)</p>                                                                                                                                                                                                                                                                                                                                                                                   |

|    |                                                                                                                                                                                                                                                                                                                          |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | c)f.reads(5)<br>d)f.readline(5)<br><b>Ans : b</b>                                                                                                                                                                                                                                                                        |
| 44 | Which function is used to force transfer of data from buffer to file?<br>a)save()<br>b)tell()<br>c)seek()<br>d)flush()<br><br><b>Ans : d</b>                                                                                                                                                                             |
| 45 | What is the output of the below program?<br><pre>def printMax(a, b):     if a &gt; b:         print(a, 'is maximum')     elif a == b:         print(a, 'is equal to', b)     else:         print(b, 'is maximum') printMax(3, 4)</pre> a) 3<br>b) 4<br>c) 4 is maximum<br>d) None of the mentioned<br><br><b>Ans : c</b> |
| 46 | Incorrect Indentation results in _____.<br>a. IndentationError<br>b. NameError<br>c. TypeError<br>d. SyntaxError<br><br><b>Ans : d</b>                                                                                                                                                                                   |
| 47 | Find the output of the following code snippet.<br><pre>for i in range(10):     if i == 5:         break     else:         print(i)</pre> a. 0 1 2 3 4<br>b. 0 1 2 3 4 5<br>c. 0 1 2 3                                                                                                                                    |

|    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | d. 1 2 3 4 5<br><br><b>Ans : a</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 48 | The syntax to write to a CSV file is<br>a. csv.DictWriter(filehandler)<br>b. csv.reader(filehandler)<br>c. csv.writer(filehandler)<br>d. csv.write(filehandler)<br><br><b>Ans : c</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 49 | What will be the output of following code?<br>def check():<br>global num<br>num=1000<br>print(num)<br>num=100<br>print(num)<br>check()<br>print(num)<br><br>a)100                      b)100                      c)1000                      d)1000<br>100                      1000                      1000                      100<br>1000                      1000                      1000                      1000<br><br><b>Ans : b</b>                                                                                                                                                                                                                                 |
|    | <b>SECTION C</b><br><b>Case Study based Questions</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|    | <b>This section consists of 6 Questions (50 -55) Attempt any 5 questions.</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|    | Ankit of class 12 is writing a program to create a CSV file “BoardDetails.csv” which will contain BoardRollNo, StuName, DateOfBirth for some entries. He has written the following code. As a programmer, help him to successfully execute the given task.<br><br><b>INCOMPLETE CODE</b><br>import _____ #Line 1<br>def addCsvFile(BoardRollNo,StuName, DateOfBirth):<br>f=open('_____', 'a', newline='') #Line 2<br>x = csv.writer(f)<br>x . .....([BoardRollNo,StuName, DateOfBirth]) #Line 3<br><br>f.close()<br><br>def readCsvFile():<br>f=open('BoardDetails.csv','__') #Line 4<br>y = csv._____ #Line 5<br>for row in y:<br>if (int(row[0])>9107750):<br>print(row[0],row[1]) |

|    |                                                                                                                                                                                                                                                                                                 |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <pre>f.close()  addCsvFile(9107752,"Sita Sharma","2003/10/21") addCsvFile(9107751,"Gita Patel","2004/11/01") addCsvFile(9107750,"Suresh Rao","2003/10/25")  readCsvFile()</pre> <p style="text-align: right;">#Line 6</p>                                                                       |
| 50 | <p>Name the module he should import in Line 1.</p> <p>a)math                      b)random                      c)binary                      d)CSV</p> <p>Ans : d</p>                                                                                                                          |
| 51 | <p>Fill in the blank in Line 2 to open the file to write/add data into the file.</p> <p>a)Shantanu.csv                      b)classxii.csv                      c) BoardDetails.csv                      d)Details.csv</p> <p>Ans : c</p>                                                       |
| 52 | <p>Choose the correct method to write the given content to the given CSV file .</p> <p>a)writer()                      b)writerow()                      c)writelines()                      d)write()</p> <p>Ans : b</p>                                                                       |
| 53 | <p>Fill in the blank in line 4 to read from the csv file.</p> <p>a)'w+'                      b)'r'                      c) 'wc'                      d)'rb'</p> <p>Ans : b</p>                                                                                                                  |
| 54 | <p>Fill in the blank in Line 5 to read data from the csv file</p> <p>a)read()                      b)readline()                      c)reader()                      d)readlines()</p> <p>Ans : c</p>                                                                                           |
| 55 | <p>Write the output he will obtain while executing Line 6.</p> <p>a) 9107750 Suresh Rao<br/>9107752 Sita Sharma</p> <p>b) 9107752 Sita Sharma<br/>9107751 Gita Patel</p> <p>c) 9107751 Gita Patel<br/>9107750 Suresh Rao</p> <p>d)9107752 Sita Sharma<br/>9107750 Suresh Rao</p> <p>Ans : b</p> |

\*\*\*\*\*