CHAPTER

3

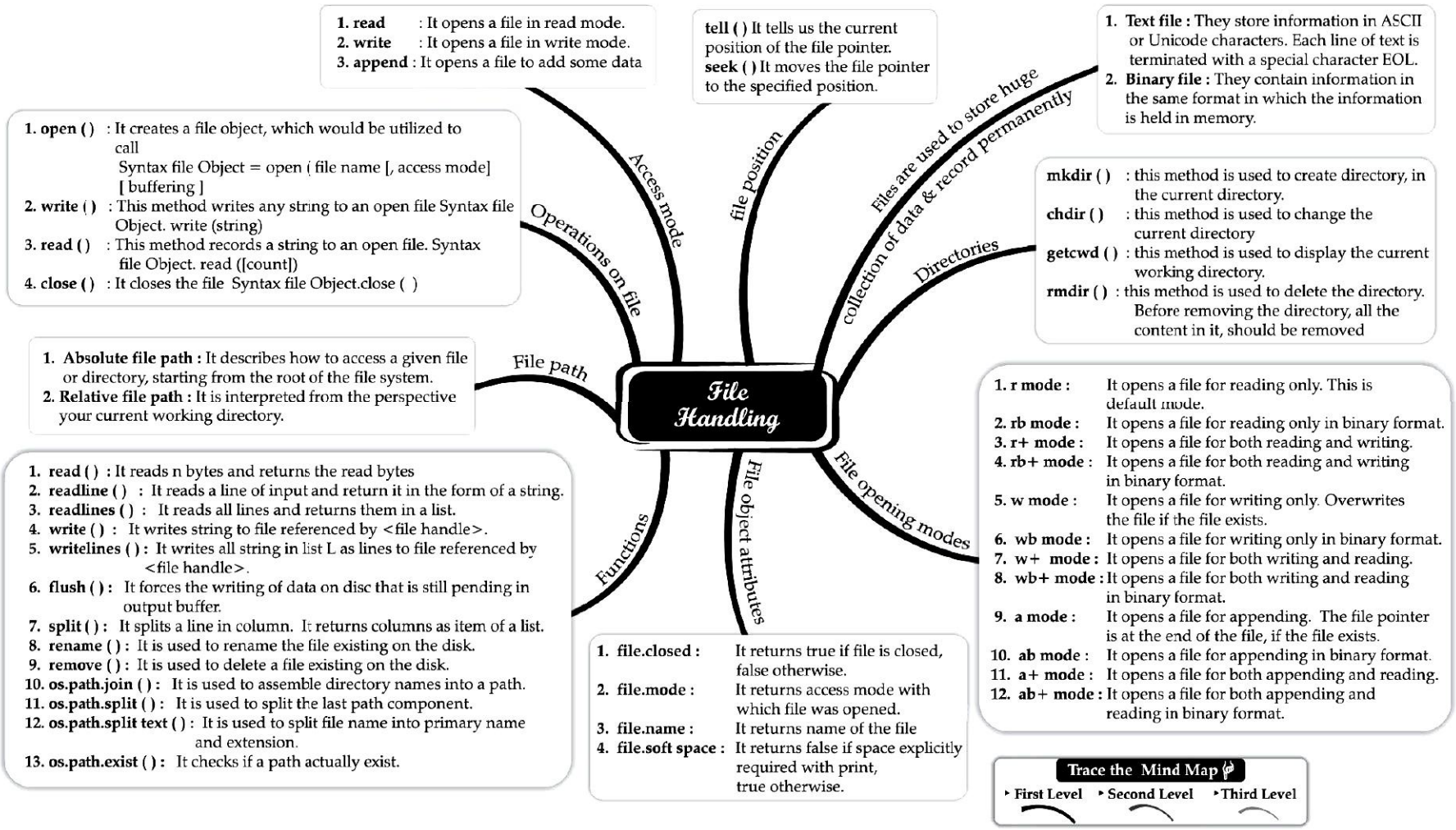
FILE HANDLING

Syllabus

- ***Introduction to files, types of files (Text file, Binary file, CSV file), relative and absolute path***
- ***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( )***

Revision Notes

- Files are used to store huge collection of data and records permanently.
- Many applications require large amount of data. In such situation, we need to use some devices such as hard disk, compact disc etc, to store the data.
- **Need for a Data File**
  - It is a convenient way to deal with large quantities of data.
  - To avoid input of data multiple times during program execution.
  - To share data between various programs.
- **Types of files**
  - **Text files** store information in ASCII or Unicode characters. In text file, each line of text is terminated, (delimited) with a special character known as EOL (End of Line) character.
  - **Binary files** are just files that contain information in the same format in which the information is held in memory, i.e., In binary file, there is no delimiter for a line.
  - **CSV (Comma Separated Value) files** are a common file format for transferring and storing data.
- Access modes specify the type of operations to be performed on the opened file.


File Handling
Access mode
1. read : It opens a file in read mode.
2. write : It opens a file in write mode.
3. append : It opens a file to add some data
file position
tell ( ) It tells us the current position of the file pointer.
seek ( ) It moves the file pointer to the specified position.
Files are used to store huge collection of data & record permanently
1. Text file : They store information in ASCII or Unicode characters. Each line of text is terminated with a special character EOL.
2. Binary file : They contain information in the same format in which the information is held in memory.
Operations on file
1. open ( ) : It creates a file object, which would be utilized to call
Syntax file Object = open (file name [, access mode] [buffering]
2. write ( ) : This method writes any string to an open file Syntax file Object. write (string)
3. read ( ) : This method records a string to an open file. Syntax file Object. read ([count])
4. close ( ) : It closes the file Syntax file Object.close ( )
Directories
mkdir ( ) : this method is used to create directory, in the current directory.
chdir ( ) : this method is used to change the current directory
getcwd ( ) : this method is used to display the current working directory.
rmdir ( ) : this method is used to delete the directory. Before removing the directory, all the content in it, should be removed
File path
1. Absolute file path : It describes how to access a given file or directory, starting from the root of the file system.
2. Relative file path : It is interpreted from the perspective your current working directory.
File opening modes
1. r mode : It opens a file for reading only. This is default mode.
2. rb mode : It opens a file for reading only in binary format.
3. r+ mode : It opens a file for both reading and writing.
4. rb+ mode : It opens a file for both reading and writing in binary format.
5. w mode : It opens a file for writing only. Overwrites the file if the file exists.
6. wb mode : It opens a file for writing only in binary format.
7. w+ mode : It opens a file for both writing and reading.
8. wb+ mode : It opens a file for both writing and reading in binary format.
9. a mode : It opens a file for appending. The file pointer is at the end of the file, if the file exists.
10. ab mode : It opens a file for appending in binary format.
11. a+ mode : It opens a file for both appending and reading.
12. ab+ mode : It opens a file for both appending and reading in binary format.
Functions
1. read ( ) : It reads n bytes and returns the read bytes
2. readline ( ) : It reads a line of input and return it in the form of a string.
3. readlines ( ) : It reads all lines and returns them in a list.
4. write ( ) : It writes string to file referenced by <file handle>.
5. writelines ( ) : It writes all string in list L as lines to file referenced by <file handle>.
6. flush ( ) : It forces the writing of data on disc that is still pending in output buffer.
7. split ( ) : It splits a line in column. It returns columns as item of a list.
8. rename ( ) : It is used to rename the file existing on the disk.
9. remove ( ) : It is used to delete a file existing on the disk.
10. os.path.join ( ) : It is used to assemble directory names into a path.
11. os.path.split ( ) : It is used to split the last path component.
12. os.path.split text ( ) : It is used to split file name into primary name and extension.
13. os.path.exist ( ) : It checks if a path actually exist.
File object attributes
1. file.closed : It returns true if file is closed, false otherwise.
2. file.mode : It returns access mode with which file was opened.
3. file.name : It returns name of the file
4. file.soft space : It returns false if space explicitly required with print, true otherwise.
Trace the Mind Map
First Level
Second Level
Third Level

- **read**(), **readline**() and **readlines**() methods are available for reading data from the file.
- **write**() and **writelines**() are used for writing data in the file.
- **pickle** module is used in serialization of data. This allows us to store data in binary form in the file.
- **dump** and **load** functions are used to write and read data from file.
- The **open()** function creates a file object which would be utilized to call other methods associated with it.

  Syntax :

  file_object=open(filename[access_mode],[buffering])

  Here is the parameter details:

  - **filename:** The file name argument is a string value that contains the name of the file that you want to access.
  - **access_mode:** The access_mode determines the mode in which the file has to be opened i.e., read, write, append, etc. A complete list of possible values is given below in the table. This is optional parameter and the default file access mode is read (r).

File Opening Modes

MODES	DESCRIPTION
r	Opens a file for reading only in text format. The file pointer is placed at the beginning of the file. This is the default mode.
rb	Opens a file for reading only in binary format. The file pointer is placed at the beginning of the file. This is the default mode.
r+	Opens a file for both reading and writing. The file pointer will be at the beginning of the file.
rb+	Opens a file for both reading and writing in binary format. The file pointer will be at the beginning of the file.
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.
wb	Opens a file for writing only in binary format. Overwrites the file if the file exists. If the file does not exist, creates a new file for writing.
w+	Opens a file for both writing and reading. Overwrites the file if the file exists. If the file does not exist, creates a new file for reading and writing.
wb+	Opens a file for both writing and reading in binary format. Overwrites the file if the file exists. If the file does not exist, creates a new file for reading and writing.
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.
ab	Opens a file for appending in binary format. 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.
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.
ab+	Opens a file for both appending and reading in binary format. 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.

- **Buffering:** If the buffering value is set to 0, no buffering will take place. If the buffering value is 1, line buffering will be performed while accessing a file. If you specify the buffering value as an integer greater than 1, then buffering action will be performed with the indicated buffer size. If negative, the buffer size is the system default (default behaviour).
- **The file object attributes:** Once a file is opened and you have one file object, you can get various information related to that file. Here is a list of all attributes related to the file object:

ATTRIBUTES	DESCRIPTION

file.closed	Returns True if file is closed, False otherwise.
file.mode	Returns access mode with which file was opened.
file.name	Returns name of the file.
file.softspace	Returns False if space explicitly required with print, True otherwise.

- **file () :** This is same as open ().
- **Random Access :** There are two functions that allow us to access a file in a non-sequential or random mode.
  - **tell() :** It tells us the position of the file pointer.
  - **seek() :** It moves the file pointer to the position specified.
- **Functions**

  (a) read () : syntax: <file handle>.read([n])

  It reads at most n bytes and returns the read bytes as string. If `n' is not specified it reads the entire file.

  (b) readline () : syntax: <file handle>.readline ([n])

  It reads a line of input, and returns it in the form of a string.

  (c) readlines () : syntax: <file handle>.readlines ()

  It reads all lines and returns them in a list.

  (d) write () : syntax: <filehandle>.write (str1)

  It writes string str1 to file referenced by <file handle>

  (e) writelines () : syntax: <file handle>.writelines (L).

  It writes all strings in list L as lines to file referenced by <file handle>

  (f) flush () : syntax: <file object>.flush()

  It forces the writing of data on disc that is still pending in output buffer.

  (g) Importing sys module lets you read/write from the standard input/output device using sys.stdin.read () and sys. stdout.write().

  (h) split () function splits a line in columns. It returns columns as items of a list.

  (i) rename () function is used to rename a file existing on the disk.

  syntax: os.remane(<current_file_name>,<new_file_name>)

  (j) remove () function is used to delete a file existing on the disk.

  syntax: os.remove(<file_name>)

  (k) os.path.join () is used to assemble directory names into a path.

  (l) os.path.split () is used to split off the last path component.

  (m)os.path.splittext() is used to split file name into primary name and extension.

  (n) os.path.exists () function checks if a path actually exists.
- **Absolute File Path :** It describes how to access a given file or directory starting from the root of the file system.
- **Relative File Path :** It is interpreted from the perspective of your current working directory.

Reading CSV files with CSV

Reading from a CSV file is done using the reader object. The CSV file is opened as a text file with Python's built in open() function, which returns a file object.

e.g.

```
import CSV
with open("Employee.txt") as CSV_file:
    CSV_reader = CSV.reader (CSV_file, delimiter = ',')
    line _count = 0
    for row in CSV_reader:
```

```
if line_count = = 0:
    print (f' column names are {",".join(row}')
    line_count + = 1
else :
    print (f '\t{row [0]} works in the {row[1]}
    department, and was born in {row[2]}.')
    line_count + = 1
print (f'Processed {line_count} lines.')
```

Optional Python CSV reader Parameters

The reader object can handle different styles of CSV files by specifying additional parameters, some of which are shown below:

- Delimiter specifies the character used to separate each field. The default is the comma (',').
- quotechar specifies the character used to surround fields that contain the delimiter character. The default is a double quote (' " ').
- escapechar specifies the character used to escape the delimiter character, in case quotes are not used. The default is no escape character.

Writing CSV Files with CSV

You can also write to a CSV file using a writer object and the write_row() methods:

e.g.

```
import CSV
    with open ('Employee_file.CSV', mode = 'w') as Employee_file:
    Employee_writer = CSV.writer (Employee_file, delimiter
    = ',' quotechar = ' " ' quoting = CSV. Quote_Minimal)
    Employee_writer.writerow(['Rahul' , Manager' , 'April'])
    Employee_writer.writerow(['Neha' , 'IT' , 'June'])
```

Know the Terms

- CSV stands for Comma Separated Values.
- Pickle module can be used to store any kind of object in file as it allows us to store Python objects with their structure.
- File Handle serve as a link to a file residing on the computer.
- File Mode governs the type of operations possible in the operand file. The default mode is read ('r')
- flush () function forces the writing of data on disc still pending in output buffers.
- seek () method can be used to position the file object at particular place in the file.
- tell () method returns an integer giving the current position of file pointer in the file.

STAND ALONE MCQs (1 Mark each)

Q. 1. To open a file c:\scores.txt for reading, we use _____

(A) infile = open("c:\scores.txt", "r")

(B) infile = open("c:\\scores.txt", "r")

(C) infile = open(file = "c:\scores.txt", "r")

(D) infile = open(file = "c:\\scores.txt", "r")

Ans. Option (B) is correct.

Explanation: Open () function creates a file object which would be utilised to call other methods associated with it.
file object = open [file name (acess_mode]), (buffering)] open (file name, [acess_mode] , [buffering])

Q. 2. Which of the following statements are true?

(A) When you open a file for reading, if the file does not exist, an error occurs

(B) When you open a file for writing, if the file does not exist, a new file is created

(C) When you open a file for writing, if the file exists, the existing file is overwritten with the new file

(D) All of the mentioned

Ans. Option (D) is correct.

Explanation: Given all options are true, when we open a file for recording, if the file does not exist, an error occurs. When we open a file for writing, if the file does not exist, a new file is created.

When we open a file for writing, if the file exists, the existing file is overwritten with the new file.

Q. 3. To read two bytes from a file object infile, we use _____

(A) infile.read(2)

(B) infile.read()

(C) infile.readline()

(D) infile.readlines()

Ans. Option (A) is correct.

Explanation: read () function reads at most n bytes and returns the read bytes as string.
Syntax <file-handle>.read [[n]] read ([n])
to read two character, so n = 2 and file name infile
infile.read (2)

Q. 4. To read the entire remaining contents of the file as a string from a file object infile, we use _____

(A) infile.read(2)

(B) infile.read()

(C) infile.readline()

(D) infile.readlines()

Ans. Option (B) is correct.

Explanation: In read () function, if n is not specified then it reads entire file.

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

```
f = None
for i in range (5):
  with open("data.txt", "w") as f:
   if i > 2:
        break
   print(f.closed)
```

(A) True **(B)** False

(C) None **(D)** Error

Ans. Option (A) is correct.

Explanation: It will give output as true.

Q. 6. Which function is used to split a line in columns?

(A) split() **(B)** spt()

(C) split_line() **(D)** All of these

Ans. Option (A) is correct.

Explanation: split () function splits a line in columns. It returns columns as item of a list.

Q. 7. It determines the mode in which the file has to be opened.

(A) file name **(B)** access_mode

(C) dump **(D)** pickle

Ans. Option (B) is correct.

Explanation: The access_mode determine the mode in which the file has to be opened i.e., read, write, append, etc.

Q. 8. Text files store information in _____ characters.

(A) ASCII **(B)** Unicode

(C) (A) and (B) **(D)** None of these

Ans. Option (C) is correct.

Explanation: Text files store information in ASCII or unicode characters. In text file, each line of text is terminated, with a special character known as EOL (End of line) character.

ASSERTION AND REASON BASED MCQs (1 Mark each)

Directions : In the following questions, A statement of Assertion (A) is followed by a statement of Reason (R). Mark the correct choice as.

(A) Both A and R are true and R is the correct explanation for A.

(B) Both A and R are true and R is not correct explanation for A.

(C) A is true but R is false.

(D) A is false but R is true.

Q. 1. Assertion (A): CSV stands for comma separated value.
Reason (R): CSV files are a common file format for transferring and storing data.

Ans. Option (B) is correct.

Explanation: CSV stands for comma separated value. These files are common file format for transferring and storing data. The ability to read, manipulate and write data to and from CSV files using Python is a key skill to master for any data scientist or business analysis.

Q. 2. Assertion (A): Access mode 'a' opens a file for appending.
Reason (R): The file pointer is at the end of the file if the file exists

Ans. Option (A) is correct.

Explanation: Acess mode '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.

Q. 3. Assertion (A): Text file stores information in ASCII or unicode characters.
Reason (R): In text file, thee is no delimiter for a line.

Ans. Option (C) is correct.

Explanation: Text file store information in ASCII or unicode character. IN text file, each line of text is terminated (determined with a special character known as EOL (end of line) character.

CASE-BASED MCQs

Attempt any four sub parts from each question. Each sub part carries 1 mark.

I. Text File

A text file can be understood as a sequence of characters consisting of alphabets, numbers and other special symbols. When we open a text file using a text editor (e.g., Notepad), we see several lines of text. However, the file contents are not stored in such a way internally. Rather, they are stored in sequence of bytes consisting of 0s and 1s. In ASCII, UNICODE or any other encoding scheme, the value of each ASCII value and shows us the equivalent character that is readable by the human being. For example, the ASCII value 65 (binary equivalent 1000001) will be displayed by a text editor as the letter 'A' since the number 65 in ASCII character set represents 'A'. Each line of a text files is terminated by a special character as EOL. However, other characters can be used to indicate EOL. When a text editor or a program interpreter encounters the ASCII equivalent of the EOL character, it displays the remaining file contents starting from a new line. Contents in a text file are usually separated by whitespace, but comma (,) and tab (\t) are also commonly used to separate values in a text file.

Q. 1. The file extension (s) used for text files is/are

(A) .txt **(B)** .py
(C) .csv **(D)** All of these

Ans. Option (D) is correct.

Explanation: Text file store information is ASCII or unicode characters. In text file, each line of text is terminated with a special character known as EOL character. The file extension used for text file is .txt.

Q. 2. What is the default EOL character in Python?

(A) \n **(B)** \t
(C) \e **(D)** \l

Ans. Option (A) is correct.

Explanation: EOL (End of line) character in Python represents by \ *n*

Q. 3. Each line of a text file is terminated by a special character called

(A) DNS **(B)** IP
(C) CSV **(D)** EOL

Ans. Option (D) is correct.

Explanation: Each line of a text file is terminated a special character called EOL (End of line) when a text editor or a program interpreter encounters the ASCII equivalent of the EOL character

Q. 4. How can you separate the content in a text file?

(A) whitespace **(B)** tab
(C) comma **(D)** All of these

Ans. Option (D) is correct.

Explanation: A text file can be sequence of characters consisting of alphabets, number and other special symbols. The content of text file can be separate by white space, tab and comma.

Q. 5. The number 65 in ASCII character set represents

(A) D **(B)** A
(C) C **(D)** B

Ans. Option (B) is correct.

Explanation: The ASCII value 65 (binary equivalent 1000001) will be displayed by a text editor as the letter A.

II. Binary Files

Binary files are also stored in terms of bytes (0s and 1s), but unlike text files, these bytes do not represent the ASCII values of characters. Rather, they represent the actual content such as image, audio, video, compressed versions of other files, executable files, etc. These files are not human readable.

Thus, trying to open a binary file using a text editor will show some garbage values.

We need specific software to read or write the contents of a binary file. Binary files are stored in a computer in a sequence of bytes. Even a single bit change can corrupt the file and make it unreadable to the supporting application. Also, it is difficult to remove any error which may occur in the binary file as the stored contents are not human readable. We can read and write both text and binary files through Python programs.

Q. 1. What is the extension of binary files?

(A) .dat **(B)** .bin
(C) .txt **(D)** .file

Ans. Option (A) is correct.

> ***Explanation:*** Binary files are also stored in terms of bytes, but unlike text file, these bytes do not represent the ASCII values of characters. The extension of binary files, is .dat.

Q. 2. What 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

Ans. Option (C) is correct.

> ***Explanation:*** Binary files represents the actual content such as image, audio, video, compressed versions of other files executable files etc. These files are not human readable. Thus, trying to open a binary file using a text editor will show some garbage values.

Q. 3. In computer, binary files are stored in terms of

(A) bit **(B)** bytes
(C) nibble **(D)** mnemonics

Ans. Option (B) is correct.

> ***Explanation:*** Binary files are also stored in terms of bytes. Byte is measurement of memory unit.

Q. 4. Binary files are human readable or not?

(A) Yes **(B)** No
(C) Depend on data **(D)** Sometimes

Ans. Option (B) is correct.

> ***Explanation:*** No, binary files are not human readable. So, if you try to open a binary file using text editor, it will show garbage values.

Q. 5. Binary files represent the actual content

(A) image **(B)** audio
(C) video **(D)** All of these

Ans. Option (D) is correct.

> ***Explanation:*** Binary files represent the actual content such as image, audio, video, compressed versions of other files, executable file etc.

III. The Pickle Module

To save any object structure along with data, Python provides a module called Pickle. The module pickle is used for serializing and de-serializing and Python object structure. Pickling is a method of preserving food items by placing them in some solution, which increases the shelf life. In other words, it is a method to store food items for later consumption. Serialization is the process of transforming data or an object in memory (RAM) to stream of bytes called byte streams. These bytes streams in a binary file can then be stored in a disk or in a database or sent through a network. Serialization process is also called pickling. De-serialization or unpickling is the inverse of pickling process where a byte stream is converted back to python object. The pickle module deals with binary files. Here, data are not written but dumped and similarly, data are not read but loaded. The pickle Module must be imported to load and dump data. The pickle module provides two methods – dump( ) and load( ) to work with binary files for pickling and unpickling, respectively.

Q. 1. _____ implements binary protocols for serializing and de-serializing a Python object structure.

(A) pickle module **(B)** unpickle module
(C) math module **(D)** random module

Ans. Option (A) is correct.

> ***Explanation:*** The pickle module is used for implementing binary protocols for serializing and de-serializing a Python object structure. The pickle module deals with binary files.

Q. 2. Which method is used to convert (pickling) python object for writing data in a binary file?

(A) load( ) **(B)** dump( )
(C) seek( ) **(D)** tell( )

Ans. Option (B) is correct.

> ***Explanation:*** dump () method is used to convert Python object for writing data in a binary file. The dump (c) needs the json file name in which the output has to be stored as an argument.

Q. 3. This method is used to load (unpickling) data from a binary file.

(A) load( ) **(B)** dump( )
(C) seek( ) **(D)** tell( )

Ans. Option (A) is correct.

> ***Explanation:*** load () method is used to load (unpickling) data from a binary file. It takes a file object and returns the json object.

Q. 4. It is the process by which a python object is converted to a byte stream.

(A) Unpickling **(B)** loading
(C) pickling **(D)** Dumping

Ans. Option (C) is correct.

> ***Explanation:*** Pickling is a process where a Python object hierarchy is converted into a byte steam. Serialization process is also called pickling.

Q. 5. The syntax of dump( ) method

(A) dump (data_object, file_object)
(B) dump [data_object, file_object]
(C) dump [data_object]
(D) dump [file_object]

Ans. Option (A) is correct.

> ***Explanation:*** dump () method is used to convert Python object for writing data in a binary file. The syntax of dump () is dump (data_object, file_object).

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

CSV File

(A) AKSHAY.XII,A
(B) ABHISHEK.XII,A
(C) ARVIND.XII,A
(D) RAVI.XII,A
(E) ASHISH.XII,A

Incomplete Code

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

Q. 1. Identify the suitable code for blank space in line marked as Statement-1.

(A) csv file **(B)** CSV
(C) csv **(D)** Csv

Ans. Option (C) is correct.

> ***Explanation:*** csv (comma separated values) file is importing in statement 1.

Q. 2. Identify the missing code for blank space in line marked as statement-2?

(A) "School.csv", "w" **(B)** "Student.csv", "w"
(C) "Student.csv", "r" **(D)** "School.csv", "r"

Ans. Option (B) is correct.

> ***Explanation:*** In statement, we need to open a file naming student CSV for writing.
> So, we will we "Student.csv", "w" in blank space.

Q. 3. Choose the function name (with argument) that should be used in the blank space of line marked as Statement-3

(A) reader(fh) **(B)** reader(MyFile)
(C) writer(fh) **(D)** writer(MyFile)

Ans. Option (C) is correct.

> ***Explanation:*** In statement 3, writer (ch) should be used. CSV, writer class to insert data to the CSV file. This class returns a writer object which is responsible for converting the user's data into a delimited string.

Q. 4. Identify the suitable code for blank space in marked as statement-4.

(A) 'ROLL_NO', 'NAME', 'CLASS', 'SECTION'
(B) ROLL_NO, NAME, CLASS, SECTION
(C) 'roll_no', 'name', 'class', 'section'
(D) roll_no,name,Class, Section

Ans. Option (D) is correct.

> ***Explanation:*** In statement 4, variable rec stores all other variables as roll-no, name class, section in list form.

Q. 5. Choose the function name that should be used in the blank space of line marked as Statement-5 to create the desired CSV File?

(A) dump( ) **(B)** load( )
(C) writerows( ) **(D)** writerow( )

Ans. Option (C) is correct.

> ***Explanation:*** In statement 5, its create a desired CSV file writerows( ) is used.
> This function writes each sequence in a list as a comma separated line of items in the file.
> This separated line of items in the file.

V. Amritya Seth is a programmer, who has recently been given a task to write a python code to perform the following binary file operations with the help of two user defined functions/modules:

[CBSE Board QB 2021]

(a) AddStudents() to create a binary file called STUDENT.DAT containing student information-roll number, name and marks (out of 100) of each student.

(b) GetStudents() to display the name and percentage of those students who have a percentage greater than 75. In case there is no student having percentage > 75 the function displays an appropriate message. The function should also display the average percent.

He has succeeded in writing partial code and has missed out certain statements, so he has left certain queries in comment lines. You as an expert of python have to provide the missing statements and other related queries based on the following code of Amritya.

Answer any four questions (out of five) from the below mentioned questions.

```
import pickle
def AddStudents ( ) :
    _____ #1 statement to open the binary file
to write data
    while True:
    Rno = int (input("Rno :"))
    Name = input ("Name : ")
    Percent = float (input ("Percent : "))
    L = [Rno, Name, Percent]
    _____ #2 statement to write the list L into
the file
    Choice = input ("enter more (y/n) : ")
        if Choice in "nN":
                    break
    F.close( )
def GetStudents ( ) :
    Total = 0
    Count rec = 0
    Count above 75 = 0
    with open ("STUDENT.DAT", "rb") as F :
while True:
                    try :
                                _____ #3 statement
to read from the file
                                        Count rec+=1
                                        Total+=R [2]
                                        if R [2] > 75 :


print (R [1], "has percent = ", R [2])
                                Count above 75+=1
    except:
    break
if Count above 75==0:
                                print ("There is no
student who has percentage more than 75")
    average=Total/Count rec
                                print ("average percent
of class = ", average)
    AddStudents ( )
    GetStudents ( )
```

Q. 1. Which of the following commands is used to open the file "STUDENT.DAT" for writing only in binary format? (marked as # 1 in the Python code)

(A) F= open("STUDENT.DAT",'wb')
(B) F= open("STUDENT.DAT",'w')
(C) F= open("STUDENT.DAT",'wb+')
(D) F= open("STUDENT.DAT",'w+')

Ans. Option (A) is correct.

> ***Explanation:*** To open the file named "STUDENT .DAT" for writing only in binary format, we should used open () function and '*wb*' acess mode.
> F = open ("student.dat", '*wb*')

Q. 2. Which of the following commands is used to write the list L into the binary file, STUDENT.DAT? (marked as #2 in the Python code)

(A) pickle.write(L,f)
(B) pickle.write(f, L)
(C) pickle.dump(L,F)
(D) f=pickle.dump(L)

Ans. Option (C) is correct.

> ***Explanation:*** dump () method is used to convert Python object for writing data in a binary file.
> dump (data-object, file object)
> In given question, data-object is L and file-object is F, so pickle. instead of, dump (L, F)

Q. 3. Which of the following commands is used to read each record from the binary file STUDENT.DAT? (marked as #3 in the Python code)

(A) R = pickle.load(F)
(B) pickle.read(r,f)
(C) r=pickle.read(f)
(D) pickle.load(r,f)

Ans. Option (A) is correct.

> ***Explanation:*** load () method is used to load data from a binary file. To read each record from the binary file with file object F, use
> R = pickle. pickle.load(F)

Q. 4. 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.

Ans. Option (A) is correct.

> ***Explanation:*** 'It' opens a file for both reading and writing. The file pointer will be at the beginning of the file.
> ' *wb*' opens a file for both writing and reading. Overwrites the file if the file exists.
> ' *a*' opens a file for appending. The file pointer is at the end of the file if the file exists.

Q. 5. 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

Ans. Option (D) is correct.

> ***Explanation:*** seek () us a built in function used for file handling. It moves the file pointer to the specified position.

VI. 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.

[CBSE QB 2021]

- 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 number 1.

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

Q. 1. In which mode Arun should open the file in Statement-1?

(A) r **(B)** r+

(C) rb **(D)** wb

Ans. Option (C) is correct.

> ***Explanation:*** In statement 1, ' *rb*' made should be used to open the file *rb* opens a file for reading only in binary format. The file pointer is placed at the beginning of the file. This is default mode.

Q. 2. Identify the suitable code to be used at blank space in line marked as Statement-2

(A) if(rec[0]==1) **(B)** for i in range(10)

(C) try **(D)** pass

Ans. Option (C) is correct.

> ***Explanation:*** In statement 2, try will be used in blank as any error or exception occur.

Q. 3. 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)

Ans. Option (C) is correct.

> ***Explanation:*** In statement 3, load (*f*) will be used where load (file-object) is used to load data from a binary file.

Q. 4. What will be the suitable code for blank space in line marked as Statement-4.

(A) rec[0]==2 **(B)** rec[1]==2

(C) rec[2]==2 **(D)** rec[0]==1

Ans. Option (D) is correct.

> ***Explanation:*** The statement 4, there are condition rec [0]== 1 used if statement. If this condition becomes true then print (rec) will be execute.

Q. 5. Which statement Arun should use at blank space in line marked as Statement- 5 to close the file.

(A) file.close( ) **(B)** close(file)

(C) f.close( ) **(D)** close( )

Ans. Option (C) is correct.

> ***Explanation:*** The statement 5, to close the file with file object *f*, f.close () is used where close () method is used to close the opened file.

VII. Radha Shah is a programmer, who has recently been given a task to write a python code to perform the following CSV file operations with the help of two user defined functions/modules:

[CBSE QB 2021]

(a) CSVOpen( ) : to create a CSV file called BOOKS. CSV in append mode containing information of books – Title, Author and Price.

(b) CSVRead( ) : to display the records from the CSV file called BOOKS.CSV where the field title starts with 'R' .

She has succeeded in writing partial code and has missed out certain statements, so she has left certain queries in comment lines.

```
import csv
def CSVOpen( ) :
```

```
    with open ('books.csv',' _____ ', newline=' ')
    as csvf :                              #Statement-1
    cw= _____                              #Statement-2
    _____                    #Statement-3
    Cw.writerow (['Rapunzel', 'Jack', 300])
    Cw.writerow (['Barbie', 'Doll', 900])
    Cw.writerow (['Johnny', 'Jane', 280])
def CSVRead ( ) :
    try :
    with open ('books.csv', 'r') as csvf :
            cr=
                                           #Statement-4
             for r in cr :
              if _____ :  #Statement-5
                          print (r)
                except :
    print ('File Not Found')
CSVOpen ( )
CSVRead ( )
```

You as an expert of Python have to provide the missing statements and other related queries based on the following code of Radha.

Answer any four questions (out of five) from the below mentioned questions.

Q. 1. Choose the appropriate mode in which the file is to be opened in append mode (Statement 1)

(A) w+ **(B)** ab

(C) r+ **(D)** a

Ans. Option (D) is correct.

> ***Explanation:*** ' *a*' access made opens a file for appending. The file painter 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 ceates a new file for writing.

Q. 2. Which statement will be used to create a csv writer object in Statement 2.

(A) csv.writer(csvf) **(B)** csv.writer(csvf)

(C) csvf.writer( ) **(D)** cs.writer(csvf)

Ans. Option (B) is correct.

> ***Explanation:*** In statement 2, CSV, writer (CSVf) is used. CSV. writer class is used to insert data to the CSV file. This class returns a writer object which is responsible for converting the user's into a delimited string.

Q. 3. Choose the correct option for Statement 3 to write the names of the column headings in the CSV file, BOOKS.CSV.

(A) cw.writerow('Title', 'Author', 'Price')

(B) cw.writerow(['Title', 'Author', 'Price'])

(C) cw.writerows('Title', 'Author', 'Price')

(D) cw.writerows(['Title', 'Author', 'Price'])

Ans. Option (B) is correct.

> ***Explanation:*** In statement 3, to write the names of column heading or one row into a file, use writerow () method writes a single row at a time. Field row can be written using this method.

Q. 4. Which statement will be used to read a csv file in Statement 4.

(A) cs.read(csvf) **(B)** csv.reader(csvf)

(C) csvf.read( ) **(D)** csvf.reader(cs)

Ans. Option (B) is correct.

> ***Explanation:*** In statement 4, reader (CSVf) will be used to read a CSV file.
> reader () used to read the file which returns an iterable reader object.

Q. 5. Fill in the appropriate statement to check the field Title starting with 'R' for Statement 5 in the above program.

(A) r[0][0]=='R' **(B)** r[1][0]=='R'

(C) r[0][1]=='R' **(D)** r[1][1]=='R'

Ans. Option (A) is correct.

> ***Explanation:*** In statement 5, there is condition *r*[0] [0] == 'R' used to check the field. If this condition is true, print (*r*) statement will be executed.

VIII. 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 less than 4 characters. Your teacher intentionally kept few blanks in between the code and asked you to fill the blanks so that code will run to find desired result. Do the needful with the following python code. **[CBSE QB 2021]**

```
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 ( )
```

Q. 1. Write mode of opening the file in statement-1?

(A) a **(B)** ab

(C) w **(D)** r

Ans. Option (D) is correct.

> ***Explanation:*** *r* mode opens a file for reding only in text format. The file painter is placed at the beginning of the file.

Q. 2. Fill in the blank in statement-2 to read the data from the file.

(A) File.Read( ) **(B)** file.read( )

(C) read.lines( ) **(D)** readlines( )

Ans. Option (B) is correct.

> ***Explanation:*** In statement 2, file read () is used. read () reads atmost *n* bytes and returns the read bytes as string. If '*n*' is not specified, it reads the entire file.

Q. 3. Fill in the blank in statement-3 to read data word by word.

(A) Line.Split() **(B)** Line.split()
(C) line.split() **(D)** split.word()

Ans. Option (C) is correct.

> ***Explanation:*** split () function is used to split a line in columns. It return columns as items of a list.

Q. 4. Fill in the blank in Statement-4, which display the word having lesser than 4 characters.

(A) len(c)==4 **(B)** len(c)<4
(C) len ( )==3 **(D)** len ( )==3

Ans. Option (B) is correct.

> ***Explanation:*** len () function is used to count the characters in a word.

Q. 5. Fill in the blank in Statement-5 to close the file.

(A) file.close( ) **(B)** File.Close( )
(C) Close( ) **(D)** end( )

Ans. Option (A) is correct.

> ***Explanation:*** to close the file.close() is used.