

FILE HANDLING

STUDY NOTES

- In Python, an error can stop the execution of a program, make it behave abnormally or allow it to execute but deliver unexpected output.
- Every, organisation or business wants to permanently store information related to its staff, business, sales, etc.
- Data can be permanently stored in Secondary devices.
- Permanently stored data can help in reusability.
- A file can be defined as a named location in the secondary memory where the information is stored.

Types of Files:

- Computers store file in binary form.
- A file is nothing but a series of bytes stored one after the other.
- **Data files:** text file and binary file are two types of data files.
- A text file consists of human readable characters whereas binary files consist of non-human readable content.

Text file:

- A text file consists of text that is sequence of characters consisting of alphabets, numbers and other special symbols.
- The file contents are stored internally in sequence of bytes consisting of 0s and 1s.
- Every line of a text file ends with a special End of Line (EOL) character.

Binary Files:

- Binary files are also stored in terms of bytes (0s and 1s).
- Trying to open a binary file using a text editor will show some garbage values.
- Contents of a binary file can be read with the help of specific software. We can read both text and binary files through Python Programs.

Opening and Closing a Text File:

- Computer programs broadly access files either to write or read data from it.
- Following operations can be performed on the file:
 - ❖ creating and opening a file
 - ❖ traversing a file
 - ❖ writing data in a file
 - ❖ reading data from a file
- Python's IO module that has functions defined for handling files.
- In Python, you can open a file using the following syntax: `file_object= open(file_name, access_mode)`
- The open function returns a file object that is assigned to file_object.
- Read and write operations can be performed on the file_object using Python's IO module.
- Important file modes are:
 - ❖ `<r>`: read only
 - ❖ `<r+>` and `<r+>` both read and write mode
 - ❖ `<w>` write mode, if file already exists contents will be overwritten
 - ❖ `<rb>` binary and read-only

- ❖ <wb+> read write and binary mode
- ❖ <a+> or <+a> append and read mode.
- ❖ <a> append mode

- Python provides a method to close the file which should be called once, after the desired operations have been performed on the file. The syntax is as follows: `file_object.close()`.
- You can open a file using “with” clause: **Syntax:** `with open (file_name, access_mode) as file_object`.

Writing to a Text File:

- In order to write anything into a file you need to open it in write or append mode.
- In order to write to a file, we first need to open the file in write or append mode.
- If we open an existing file in write mode, all its data will be erased and the file will be positioned at the beginning of the file.
- In the other; the append mode, new data will be added at the end of an already existing data.
- `write()` or `writeline()` method can be used to write data on to the file.

Reading From a Text File:

- A file can be read in “r”, “r+”, “w+”, “w” or “ a+” mode.
- You can read the contents of a file using:
 - ❖ `read()`
 - ❖ `readline()`
 - ❖ `readlines()`

Setting Offests in a File:

- A file can be accessed in a random fashion with the help of `seek()` and `tell()` method.
- The `tell()` function returns an integer that specifies the current position of the file object in the file.
 - ❖ **Syntax:** `file_object_tell()`
- The `seek()` method positions a file at a particular position in file.
 - ❖ **Syntax:** `file_object.seek(offset[, reference_point])`
 - ❖ Reference point can be:
 - 0 – beginning of the file (also default value)
 - 1 – Current position of the file
 - 2 – end of file

Pickle Module:

- Used for serializing and de-serializing any Python object structure.
- Serialization or pickling transforms data or object in memory to stream of bytes called byte stream.
- De-serialization converts byte string back to Python object.
- Pickle module deals with binary files.
- The `dump()` method is used to convert Python objects for writing data into binary files. It is used for pickling().
- The `load()` method is used for unpickling data from binary files.

QUESTION BANK

MULTIPLE CHOICE QUESTIONS

1. A file is a/an _____ .

(a) Application	(b) Location in secondary memory
(c) Register	(d) (a) and (c)
2. Data can be stored permanently on:

(a) Primary memory	(b) Secondary memory
(c) Cache	(d) Registers

3. Computers store every file as a collection of:
 - (a) Binary numbers
 - (b) Different format
 - (c) Web pages
 - (d) Memory location
4. Data file can be:
 - (a) Only binary file
 - (b) Only text file
 - (c) Both binary and text file
 - (d) Neither binary nor text file
5. What is not true for binary files?
 - (a) it is a type of data file
 - (b) It contains non-human readable characters and symbols
 - (c) Can be accessed by any program
 - (d) Requires specific programs to access its content
6. Which of the following is not an extension for text file?
 - (a) .txt
 - (b) .py
 - (c) .csv
 - (d) .bin
7. Which is not an encoding scheme for storing text file?
 - (a) ASCII
 - (b) Unicode
 - (c) UTF-8
 - (d) UFO-6
8. The value of each character of the text file is stored as:
 - (a) A bit
 - (b) Bytes
 - (c) Megabytes
 - (d) Gigabytes
9. Every line of text file terminates with a special character called _____.
 - (a) AOL
 - (b) EOL
 - (c) ELO
 - (d) ALO
10. The default EOL in Python is:
 - (a) \n
 - (b) \n
 - (c) /n
 - (d) /n\
11. When a text editor or a program interpreter encounters the ASCII equivalent of the EOL character:
 - (a) It ends the file.
 - (b) It saves the file.
 - (c) It displays the remaining file contents starting from a new line.
 - (d) It displays the remaining file contents starting from the beginning of the file.
12. Content in text file is separated by:
 - (a) White space
 - (b) ,
 - (c) \t
 - (d) All of these
13. As compared to a .docx file a text file will not contain:
 - (a) Author's name
 - (b) Page settings
 - (c) Font types
 - (d) All of these
14. Which of the following is the similarity between the text files and the binary files?
 - (a) Information stored in 0's and 1's.
 - (b) Bytes do not represent ASCII values of characters.
 - (c) Binary information represents actual content such as video, audio, etc.
 - (d) Information is not stored in human readable form.
15. If you open a binary file you will see:
 - (a) Garbage values
 - (b) Text value
 - (c) 0 and 1
 - (d) Bytes
16. Python language can be used to read and write:
 - (a) Binary files only
 - (b) Text files only
 - (c) Both binary and text file
 - (d) Neither binary nor text file
17. What is not true for a binary file?
 - (a) A specific software is required to read or write the contents of a binary file.
 - (b) Binary files are stored in a computer in a sequence of bytes.
 - (c) Even a single bit change can corrupt the file.
 - (d) It is easy to remove any error which may occur.
18. Which Python module has functions defined for handling files?
 - (a) io
 - (b) fileio
 - (c) oi
 - (d) fileoi

19. Which is the correct syntax for opening a file using Python?
 (a) `file_object= io.open(file_name, access_mode)` (b) `file_object= object.open(file_name, access_mode)`
 (c) `file_object= open(file_name, access_mode)` (d) `file_object= file.open(file_name, access_mode)`
20. File handle is a:
 (a) File object (b) Opened file
 (c) An in-built function (d) None of these
21. Which attribute informs whether a file is closed or not?
 (a) `<file.closed>` (b) `<file.mode>` (c) `<file.name>` (d) `<file.object>`
22. Which attribute informs information about how a file was opened?
 (a) `<file.closed>` (b) `<file.mode>` (c) `<file.name>` (d) `<file.object>`
23. Which attribute returns the name of the file?
 (a) `<file.closed>` (b) `<file.mode>` (c) `<file.name>` (d) `<file.object>`
24. Which of the following is not an access mode?
 Some of the other file access modes are
 (a) `<rb+>` (b) `<wb>` (c) `<w+>` (d) `<+x>`
25. The file offset position for `<r>` is the:
 (a) Beginning of the file (b) End of the file
 (c) Middle of the file (d) The point where it was left when last accessed
26. `<rb>` allows you to open file in:
 (a) Read mode (b) Rewrite mode (c) Read write mode (d) Binary read mode
27. The file offset position for `<w>` is :
 (a) Beginning of the file (b) End of the file
 (c) Middle of the file (d) The point where it was left when last accessed
28. In which of the file open mode, the new content will be overwritten on the old content?
 (a) `<r>` (b) `<w>` (c) `<a>` (d) `<rb>`
29. In the `open()` function, _____ is an optional argument also referred to as processing mode.
 (a) `file_mode` (b) `access_mode` (c) `access_name` (d) `access_closed`
30. When the access mode is not given, by default the file will open in:
 (a) append (b) read mode (c) write mode (d) read-write
31. What is not true for `myObject=open("myfile.txt", "a+")`?
 (a) `myfile.txt` is opened in append mode (b) `myfile.txt` is opened in read mode
 (c) The file object will be at the end of the file (d) None of these
32. Once you have performed the desired operations on the file using Python the best thing to do is:
 (a) Save it (b) Close it
 (c) Move it to safer location (d) Copy it
33. Which is the correct syntax for closing a file using Python?
 (a) `io.close(file_name, access_mode)` (b) `file_object.close(file_name)`
 (c) `file_object.close()` (d) `file.file_object.close()`
34. The best thing about _____ clause is that the file that is opened using this clause closes automatically once the control comes out of this clause.
 (a) `where` (b) `for` (c) `with` (d) `if`
35. To write into a text file you will have to open it in:
 (a) read mode (b) append mode (c) write mode (d) (b) and (c)

36. If we open an existing file in the _____ mode the previous data will be erased and the file object will be positioned at the beginning of the file.
 (a) read mode (b) append mode (c) write mode (d) (b) and (c)
37. In append mode, we can write data using:
 (a) write() (b) writeline (c) writelines() (d) (a) and (b)
38. For a newly created file, write() and append() method will give:
 (a) Same results (b) Different results (c) No results (d) Null output
39. Which method takes string as an argument and writes it to the text file?
 (a) close() (b) write() (c) read() (d) append()
40. On execution write() method returns:
 (a) The file object (b) The file handle
 (c) The number of characters written on to the file (d) All of these
41. While writing files '\n' is treated as a:
 (a) Single character (b) Two characters (c) No character (d) Special character
42. When _____ method is called, it writes data onto a buffer.
 (a) write() (b) read() (c) readlines() (d) close()
43. After write() method when _____ method is executes, the contents from the buffer are moved to the file located on the permanent storage.
 (a) read() (b) append() (c) close() (d) store()
44. The writelines() method can write multiple lines with the help of:
 (a) iterable object (b) list (c) tuple (d) all of the above
45. What is not a similarity between write() and writelines()?
 (a) Called in append mode (b) Returns number of characters written on the file
 (c) File offset position is beginning of a file (d) None of these
46. What would be the output of the following code:

```
myobject=open("C:/codemyfile.txt",'w')
lines = ("Hello everyone\n", "Multiple strings\n", "TUPLES")
myobject.writelines(lines)
myobject.close()
```

 (a) Runtime error (b) File created (c) Syntax error (d) File created and lines written
47. Method used for writing the Python objects in a binary file:
 (a) pickle() (b) unpickle() (c) load() (d) dump()
48. Method used for reading data from a binary file.
 (a) pickle() (b) unpickle() (c) load() (d) dump()
49. Python object is converted to a byte stream by the process known as Pickling.
 (a) True (b) False
50. To position the file object at a particular position in file, we use:
 (a) dump() (b) load() (c) seek() (d) tell()

INPUT TEXT BASED MCQs

Read the following passage and answer the following questions (51 to 54).

A text file can be understood as a sequence of characters consisting of alphabets, numbers and other special symbols. Files with extensions like .txt, .py, .csv, etc. are some examples of text files. 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 character of the text file is stored as bytes. So, while opening a text file, the text editor translates 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'.

51. How does a line terminate in text file?
 (a) '.' (b) Space (c) EOL character (d) BOF character
52. The default value of EOL character is:
 (a) \t (b) \x (c) \n (d) \v
53. Contents in text file are separated by:
 (a) Space (b) , (c) \t (d) All of these
54. What happens when a text editor or a program interpreter encounters the ASCII equivalent of the EOL character:
 (a) File closes
 (b) Data is saved
 (c) Remaining file contents are displayed starting from a new line.
 (d) None of the above.

ANSWERS									
Multiple Choice Questions									
1. (b)	2. (b)	3. (a)	4. (c)	5. (c)	6. (d)	7. (d)	8. (b)	9. (b)	10. (a)
11. (c)	12. (d)	13. (d)	14. (a)	15. (a)	16. (c)	17. (d)	18. (a)	19. (c)	20. (a)
21. (a)	22. (b)	23. (c)	24. (d)	25. (a)	26. (d)	27. (a)	28. (b)	29. (a)	30. (b)
31. (d)	32. (b)	33. (c)	34. (c)	35. (d)	36. (c)	37. (d)	38. (a)	39. (b)	40. (c)
41. (a)	42. (a)	43. (c)	44. (d)	45. (c)	46. (d)	47. (d)	48. (c)	49. (a)	50. (c)
Input Text Based MCQs									
51. (c)	52. (d)	53. (d)	54. (c)						

HINTS/EXPLANATION

1. A file can be defined as a named location in the secondary memory where the information is stored.
2. Data can be permanently stored in Secondary devices.
3. Computers store every file as a collection of binary numbers.
6. Bin is the extension for binary files.
8. The value of each character of the text file is stored as bytes.
11. 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.
12. 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.
13. Text files contain only the ASCII equivalent of the contents of the file.
14. In a binary file:
 - (a) Information stored in 0's and 1's.
 - (b) bytes do not represent ASCII values of characters.
 - (c) binary information represents actual content such as video, audio, etc.
 - (d) information is not stored in human readable form.

All points are true for binary file. Only point a. is true for text file.
16. Python can be used to read and write both binary and text files.

17. It is difficult to fix errors in a binary file as the stored contents are not in human readable form.
19. The correct syntax for opening a file using Python is `file_object= open(file_name, access_mode)`.
20. File handle is a file object on which file operations are performed.
22. You will come to know about the access mode in which file is opened by the `<file.mode>` option.
23. `<file.name>` returns the name of the file.
24. There is no access mode by the name `<+x>`.
25. The file offset for the file mode `<r>` is beginning of the file.
26. `<rb>` stands for binary read mode.
28. `<w>` opens the file in write mode. If the file does not exist it will be created. If it exists then all the contents will be overwritten.
29. In the `open()` function, `access_mode` is an optional argument that represents the mode in which file has to be accessed.
31. The following is true for `myObject=open("myfile.txt", "a+")`
 - (a) `myfile.txt` is opened in append mode
 - (b) `myfile.txt` is opened in read mode
 - (c) The file object will be at the end of the file
32. After the read/write operations have been performed on the file using Python. It is better to close it.
34. The best thing about with clause is that the file that is opened using this clause closes automatically once the control comes out of this clause.
35. In order to write anything into a file, you need to open it in write or append mode.
36. If we open an existing file in the write mode then all the data of the file will be erased and the file object will be positioned at the beginning of the file.
38. For a newly created file, `write()` and `append()` method will give same result. If we open an existing file in write mode, the previous data will be erased, and the file object will be positioned at the beginning of the file. On the other hand, in append mode, new data will be added at the end of the previous data but a newly created file has no data so it does not matter whether we use `write()` or `append`.
42. When `write()` method is called, it writes data onto a buffer.
43. After `write()` method when the `close()` method is executed, the contents from the buffer are moved to the file located on the permanent storage.
44. It is important to pass an iterable object like lists, tuple etc. containing strings to the `writelines()` method.
45. `Writelines()` does not return number characters written on the file.