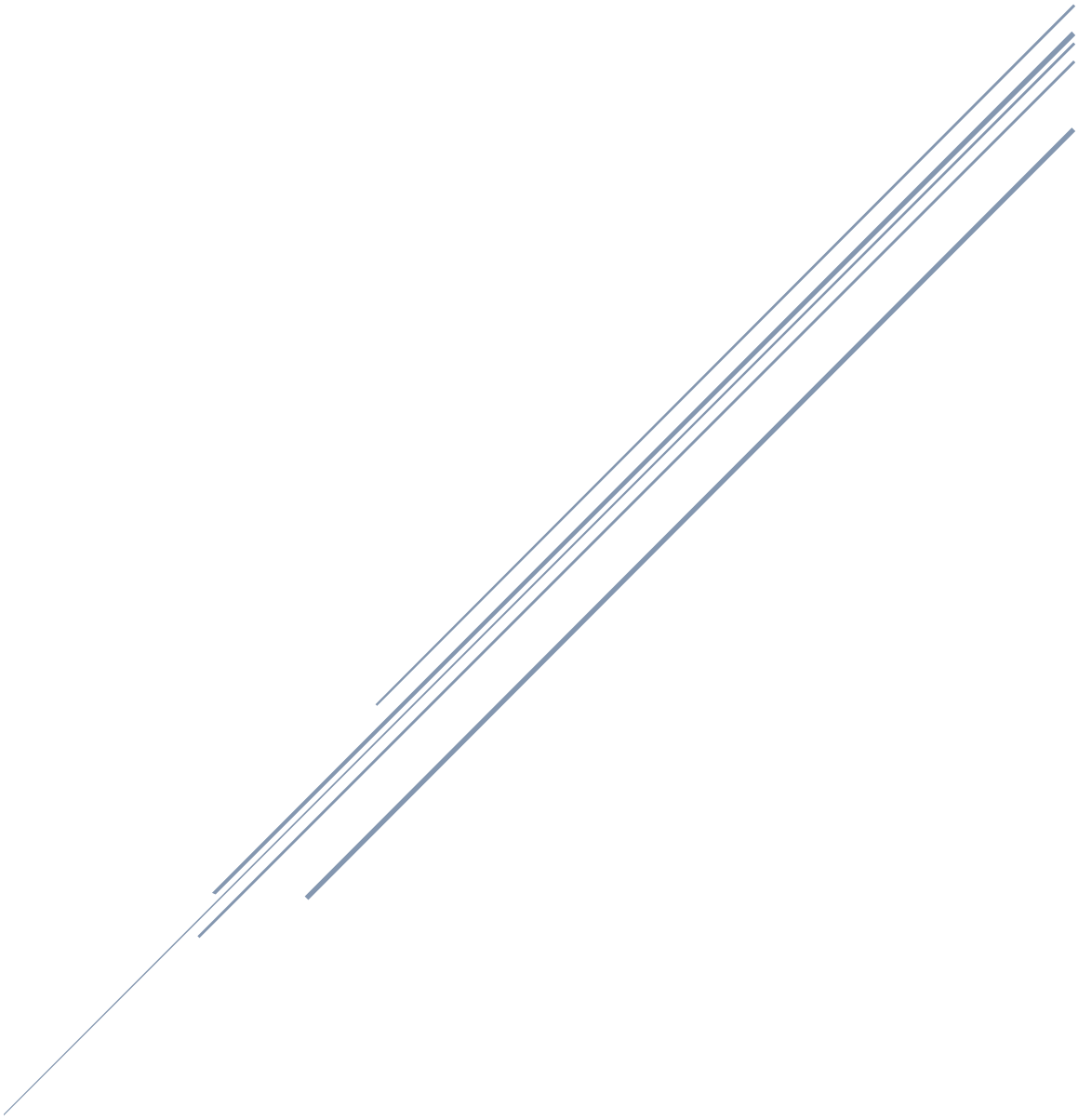
Name – Hitesh Shashikant Mhatre.

Topic: File Handling Theory Questions



Batch :2022-8935
Enrollment Number :EBEON0223761175

1. What is file handling?

- File handling in Python is simplified with built-in methods, which include creating, opening, and closing files.
- We need file handling, to store data permanently in file, so that we **can retrieve** data whenever required.
- In file Handling, Data is stored in **non-volatile memory**.
- file handling can be used to store small data.

2. Can you explain the difference modes of opening a file?

There are many modes for opening a file:

- **r - Open a file in read mode.**
- **w - Opens or create a text file in write mode.**
- **a - opens a file in append mode.**
- **r+ - opens a file in both read and write mode.**
- **a+** - opens a file in both read and write mode.
- **w+ - opens a file in both read and write mode.**

3. How do you create a text file?

The text file can be create in python with the use of built in function "Open ()" with the write mode."w". Syntax:
- f = open ('file_name','w')

4. How to read and write to an existing file?

To read and write to an existing file, a built in function can be used. Open a file with r+ mode.
r+ - opens a file in both read and write mode. Syntax: - Syntax:- f = open('file_name','r+')

5. What are some important methods used for reading from a file?

There are three types of method in reading.

Read() function

- This method is used to read used to read data/content from a file and returns a retunes ir as a string in text binary mode.
- Syntax: file_handler.read(size)

Readline() Function

- This method is used to read single line from a file.
- Syntax: file_handler.readline(Size)

Readlines() Function

- This method is used to read all the lines from a file and list of lines. mode. It retunes bytes in
- Syntax:- file_handler.readlines()

6. What are some common errors that can occur while working with files?

- FileNotFoundError: This error occurs when you try to open a file that doesn't exist.
- PermissionError: This error occurs when you try to access a file that you don't have permission to access.
- IOError: This error occurs when there is an issue with reading or writing to a file, such as if the file is opened in the wrong mode.

- ValueError: This error occurs when the data being written to a file is not in the expected format, such as trying to write a string to a file that only accepts integers.
- UnicodeDecodeError: This error occurs when there is an issue with decoding the contents of a file that contains non-ASCII characters.
- EOFError: This error occurs when you try to read beyond the end of a file.

7. What is difference between text and binary files?

Ans. Text files and binary files are two different types of computer files.

Text files contain human-readable characters, such as letters, numbers, and symbols, and are typically encoded in a format such as ASCII or Unicode. They can be edited using a simple text editor like Notepad, and their contents can be easily viewed and manipulated by humans.

Binary files, on the other hand, contain data in a non-human-readable format, such as 0s and 1s, and are typically used to store machine code, images, audio, or video data. They cannot be edited using a simple text editor, and their contents may not be easily understood or manipulated by humans.

One major difference between text files and binary files is the way they handle line breaks. In a text file, a line break is typically represented as a single newline character (**\n**). In a binary file, however, line breaks may be represented in a different way, or they may not be present at all.

8. Which function allow us to check if we have reached the end of a file?

EOF stands **for End of File** allow us to check if we have reached the end of a file

9. List down the steps involved in a processing a large file?

1. Open a file
2. Read or Write (perform operations)
3. Append
4. Close the file

10. What is the difference between write and append mode?

Ans.

Write mode (**'w'**) is used to open a file for writing, and if the file already exists, it will be overwritten. When you write to a file in write mode, any existing contents of the file are deleted, and new data is written to the file from the beginning. If the file does not exist, it will be created.
Append mode (**'a'**) is used to open a file for writing, but unlike write mode, it does not overwrite the existing contents of the file. Instead, new data is written to the end of the file. If the file does not exist, it will be created.

11. **What is the difference between read() ad read(n) functions?** Ans. **read()** is used to read the entire contents of a file as a single string. When you call **read()** on a file object, it returns a string containing all the data in the file, starting from the current file pointer position to the end of the file. **read(n)** is used to read a specific number of characters (n) from a file. When you call **read(n)** on a file object, it returns a string containing the next n characters from the file, starting from the current file pointer position

12. Differentiate between absolute pathnames and relative pathnames?

An absolute pathname is the full path of a file or directory, starting from the root directory of the file system. In other words, an absolute pathname specifies the complete path of a file or directory, starting from the top-level directory of the file system.

13. Differentiate between file modes r+ and w+ with respect to python?

r+ mode:- The **r+** mode opens a file for both reading and writing. When you open a file in **r+** mode, the file pointer is initially positioned at the beginning of the file, **w+ mode** :- opens a file for both reading and writing, but with the additional feature that it truncates the file to zero length before writing. When you open a file in **w+** mode, the file pointer is initially positioned at the beginning of the file, and you can read the file contents using the **read()** function. You can also write to the file using the **write()** function, and any existing data in the file will be overwritten

14. What is file mode? Name the default file mode

In Python, a file mode is a string that specifies the way a file should be opened. The file mode is passed as the second argument to the **open()** function.
The default file mode in Python is **"r"**, which is used to open a file for reading. If you do not specify a file mode when opening a file, Python will default to **"r"** mode. Here are some of the most commonly used file modes in Python:

- **"r"**: open the file for reading (default)
- **"w"**: open the file for writing, truncating the file first
- **"a"**: open the file for writing, but append to the end of the file rather than truncating it
- **"x"**: create a new file and open it for writing (raises an error if the file already exists)
- **"b"**: open the file in binary mode
- **"t"**: open the file in text mode (default)