**Name**: Badgujar janavi Vijay

***Topic: File Handling Theory Questions***

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 “Opne()” 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 methods in reading.

|  |  |  |
| --- | --- | --- |
| **Read() function** | **Readline() Function** | **Readlines() Function** |
| • This method is used  to read data/content  from a file and  retunes ir as a string  in text mode.  It retunes bytes in  binary mode.  • Syntax:  file\_handler.read(size)  • Size: - it represents  the number of  characters to be read  in text mode. When  using binary mode,  no need to give size. | • This method is used to  read single line from a file.    • Syntax:  file\_handler.readline(Size) | • This method is used to read  all the lines from a file and  returns a list of lines.  • 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?**

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Text Files** | **Binary Files** |
| **Definition** | A text file consists of **human**  **readable characters**, which  can be opened by any text  editor. | Binary files are made up of  **non-human readable**  **characters and symbols**, which  require specific programs to  access its contents. |
| **Form** | Stores data in form of  **characters**. It is used to store  **data and string**. | Stores data in the form of **bytes**  (group of 8 bits) |
| **Example** | Txt., JSON (JavaScript Object  Notation) | Audio, Text, Image, PDF. |

**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?**

There are 4 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?**

The difference between write and append mode are list down in the below table

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Write Mode** | **Append Mode** |
| **Existing File** | When a file is opened in write  mode ('w'), any existing contents  of the file are overwritten with  new data. | when a file is opened in append  mode ('a'), new data is added to  the end of the file without  overwriting any existing data |
| **New File** | If the file doesn't exist, a new file  is created. This means that if you  open a file in write mode and  write to it multiple times, only  the last write operation will be  reflected in the file. | If the file doesn't exist, a new file  is created. This means that if you  open a file in append mode and  write to it multiple times, each  write operation will be appended  to the end of the file. |

**11. What is the difference between read() ad read(n) functions?**

The difference between read() ad read(n) functions are list down in the below

|  |  |
| --- | --- |
| **Read()** | **Read(n)** |
| The read() method reads the entire file contents  as a single string. | The read(n) method reads the next n characters  (or bytes) of the file and returns them as a string. |

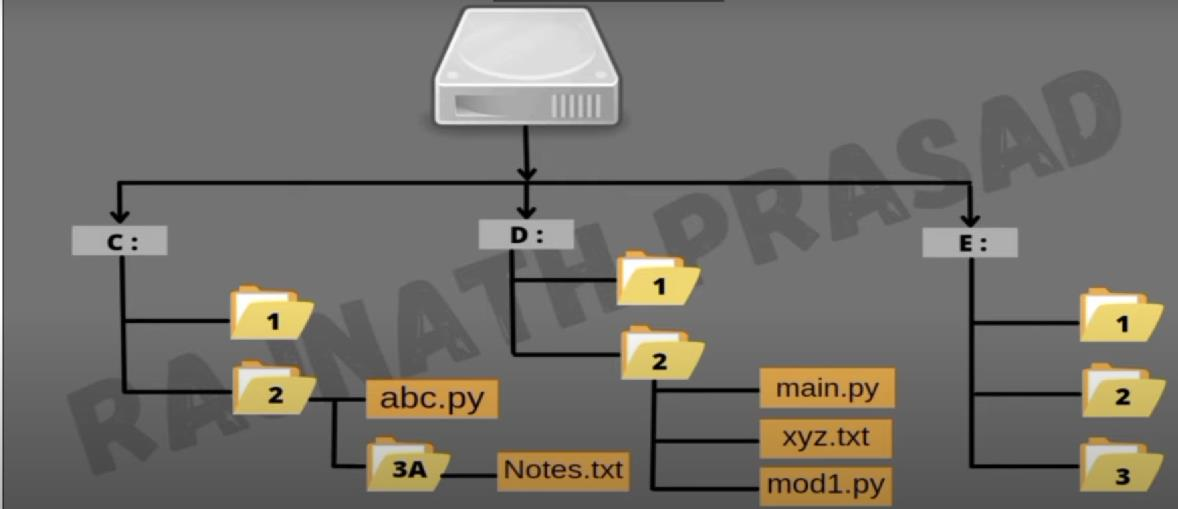
**12. Differentiate between absolute pathnames and relative pathnames?**

The difference between absolute pathnames and relative pathnames are list down in the below

table:-

|  |  |
| --- | --- |
| **Absolute Pathnames** | **Relative Pathnames** |
| It points to a specific location in the file system,  irrespective of the current working directory. It is  also called as full path or file path. | It points to the location of a directory using  current directory as a reference. It is called as  non-absolute path. |
| It refers to the location of a file or directory  (filesystem) relative to the root directory in Linux. | It refers to the location of a file or directory  (filesystem) relative to the current directory. |

Example:



|  |  |
| --- | --- |
| To access Notes, the absolute path will be  “C:\2\3A\notes.txt” | To access Notes, the relative path will be  “3A\notes.txt” |
|  |  |

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

The difference between file modes r+ and w+ are list down in the below table:-

|  |  |  |
| --- | --- | --- |
| **Parameters** | **r+ Mode** | **w+ Mode** |
| **Existing File** | When you open a file in r+  mode, the file pointer is placed  at the beginning of the file. This  means that if you start writing  to the file, it will overwrite the  existing data from the beginning  of the file. You can use the  seek() method to move the file  pointer to a specific location in  the file before writing | When you open a file in w+  mode, the file pointer is placed  at the beginning of the file, just  like in r+ mode. This means that  if you start writing to the file, it  will overwrite the existing data  from the beginning of the file.  However, since the file was  truncated to zero length when it  was opened, there is no existing  data to overwrite. You can use  the seek() method to move the  file pointer to a specific location  in the file before writing. |
| **New File** | The r+ mode opens a file for  reading and writing. If the file  does not exist, it will raise an  error. | the w+ mode opens a file for  reading and writing as well, but  it also truncates the file to zero  length. If the file does not exist,  it creates a new file |

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

• The mode determines where the file is positioned when opened, and what functions are

allowed. After you close a file, you can reopen the file in a different mode, depending on

what you are doing.

• The default file mode is read() Mode.