Module 10: Python Files

File:

- ❑ File is a named location on disk to store related information.
- ❑ It is used to permanently store data in a non-volatile memory (e.g. hard disk).

File Types:

- ❑ Python can handle two types of files:
  - ❑ 1) Text files
  - ❑ 2) Binary files

File Operations:

- ❑ It takes place in the following order:
  - ❑ Open a file.
  - ❑ Perform file operation (read/write/append)
  - ❑ Close the file.

Opening a File:

f = open("filename", "mode")

open() ➔ this functions opens a file and returns a file object(file handle).

mode:

w ➔ write data into file.

- ❑ It creates new file and stores the data.
- ❑ If the file already exists then it would overwrite the data.

r ➔ read data from the file.

- ❑ It reads the data from exisiting file
- ❑ If the file does not exists then it throws FileNotFoundError

a➔ append data to file.

- ❑ It appends the data to the end of the file.
- ❑ If the file does not exists then it creates new file.

+ in suffix if we want to read and write/append.

b in suffix if we want to work with binary data

File Operations:

Write Operations:

- ❑ **write(s)** ➔ write string s to the file and return the number of characters written.
    - ❑ Note 1: string can be text or binary data.
    - ❑ Note 2: write method does not add a newline character to the end of the string.
- ❑ **writelines():** ➔ it writes a List of string elements in the file.

Read Operations:

- ❑ **read()** ➔ returns the read bytes in the form of string.
    - ❑ If n is specified then read atmost n characters from the file.
- ❑ **readline()** ➔ reads and returns one line from the file.
- ❑ **readlines()** ➔ reads and return a list of lines from the file.

Closing a File:

- ❑ **close()** will flush out any unwritten information and close the file, after which no more writing can be done.
- ❑ On closed file any operation would lead to exception: ValueError: I/O operation on closed file.
- ❑ If an exception occurs when we are performing some operation with file, the code exits without closing the file.
- ❑ Two Options for safe closure:
    - ❑ Use try..finally block or use with statement

os module:

- ❑ The os module provides a big range of useful methods to manipulate files and directories:
    - ❑ **remove():** ➔ it is used to delete file.
    - ❑ **rename():**➔ it is used for renaming the current file name with new file name.
- ❑ The os.path is a module which provides ways of manipulating paths:
    - ❑ **isfile():** ➔ return True if the file exists.
    - ❑ **exists():** ➔ returns True if dir/file exists.