Objective: The file operations are very important as they help in working with the different data sets. This worksheet introduces few methods to read a file and write into a file using Python functions. The functions introduced in this worksheet opens a file in text or binary format. Going forward in other worksheets, the file operations will also be performed using Python packages for different file formats (.xlsx, .csv, .json etc.).

Reading from a File

- A file can be opened for reading using the ***open()*** function. It takes two necessary arguments; the first one is the file name with complete path and the second one is mode in which the file is to be opened.
- The file path should use forward slash (/) even if the Windows operating system is used. There are many values the mode argument can take. This worksheet focuses only reading ("r") and writing ("w") modes.
- A file ***sampleData.txt*** is provided along with this worksheet. Participants can use this file to experiment.
- The code snippet below opens the file in reading mode:

```
f = open("C:/Users/BITS-PC/Desktop/sampleData.txt", "r")
```

```
print(f)
```

```
<_io.TextIOWrapper name='C:/Users/BITS-PC/Desktop/sampleData.txt' mode='r' encoding='cp1252'>
```

- In the above code, the variable ***f*** is a handler (or stream) which will be used to operate on the file. When its value is printed, the output shows its name, the mode in which it is opened and encoding (cp1252 is a Latin alphabet encoding where each character is represented in a single byte. It is comparable with UTF-8 encoding.)
- Once the file handler (***f***) is available, it can be used to read the file's content using the ***read()*** function. The file in this example contains the classification data, where values are comma separated. All the content of the file is read in one go.

```
output = f.read()
print(output)
```

```
Cost,Maintenance,Doors,Persons,Boot,Safety,Class

VeryHigh,VeryHigh,2,2,Small,Low,Unacceptable
VeryHigh,VeryHigh,2,2,Small,Medium,Unacceptable
VeryHigh,VeryHigh,2,2,Small,High,Unacceptable
VeryHigh,VeryHigh,2,2,Avg,Low,Unacceptable
VeryHigh,VeryHigh,2,2,Avg,Medium,Unacceptable
VeryHigh,VeryHigh,2,2,Avg,High,Unacceptable
VeryHigh,VeryHigh,2,2,Big,Low,Unacceptable
VeryHigh,VeryHigh,2,2,Big,Medium,Unacceptable
VeryHigh,VeryHigh,2,2,Big,High,Unacceptable
VeryHigh,VeryHigh,2,4,Small,Low,Unacceptable
VeryHigh,VeryHigh,2,4,Small,Medium,Unacceptable
```

- Another way the same file can be read *line-by-line* is shown below. It uses the function ***readline().*** Can you tell the reason why the file was opened again? If the same ***readline()*** function was used in the sequence to the last example, what would happen?

```
f = open("C:/Users/BITS-PC/Desktop/sampleData.txt", "r")
nextLine = f.readline()
print(nextLine)
```

```
Cost,Maintenance,Doors,Persons,Boot,Safety,Class
```

```
nextLine = f.readline()
print(nextLine)
```

```

```

```
nextLine = f.readline()
print(nextLine)
```

```
VeryHigh,VeryHigh,2,2,Small,Low,Unacceptable
```

- Python and many other languages use **'\n'** as the new line character. The function ***readlines***(): the plural form of ***readline()*** can be used to read the content of the file in the list data type. Each element in the list is identified wherever the line break using **'\n'** takes place.

```
f = open("C:/Users/BITS-PC/Desktop/sampleData.txt", "r")
lines = f.readlines()
print(lines)
```

```
['Cost,Maintenance,Doors,Persons,Boot,Safety,Class\n', '\n', 'VeryHigh,VeryHigh,2,2,Small,Low,Unacceptable\n', 'VeryH
igh,VeryHigh,2,2,Small,Medium,Unacceptable\n', 'VeryHigh,VeryHigh,2,2,Small,High,Unacceptable\n', 'VeryHigh,VeryHigh,
2,2,Avg,Low,Unacceptable\n', 'VeryHigh,VeryHigh,2,2,Avg,Medium,Unacceptable\n', 'VeryHigh,VeryHigh,2,2,Avg,High,Unacc
eptable\n', 'VeryHigh,VeryHigh,2,2,Big,Low,Unacceptable\n', 'VeryHigh,VeryHigh,2,2,Big,Medium,Unacceptable\n', 'VeryH
igh,VeryHigh,2,2,Big,High,Unacceptable\n', 'VeryHigh,VeryHigh,2,4,Small,Low,Unacceptable\n', 'VeryHigh,VeryHigh,2,4,S
mall,Medium,Unacceptable\n']
```

- The file which was opened previously, need to be closed explicitly using the ***close()*** function. If this is not done the file may not be utilized for other purpose by the operating system. You can verify it by moving the opened file to some other directory. It will not happen in Windows.

```
f.close()
```

- It is cumbersome to remember which files are opened and to close them in the end in the large code. This can be managed using a different approach as shown in the example below:

```
with open ("C:/Users/BITS-PC/Desktop/sampleData.txt", "r") as f:
    output = f.read()
    print(output)
```

```
Cost,Maintenance,Doors,Persons,Boot,Safety,Class

VeryHigh,VeryHigh,2,2,Small,Low,Unacceptable
VeryHigh,VeryHigh,2,2,Small,Medium,Unacceptable
VeryHigh,VeryHigh,2,2,Small,High,Unacceptable
VeryHigh,VeryHigh,2,2,Avg,Low,Unacceptable
VeryHigh,VeryHigh,2,2,Avg,Medium,Unacceptable
VeryHigh,VeryHigh,2,2,Avg,High,Unacceptable
VeryHigh,VeryHigh,2,2,Big,Low,Unacceptable
VeryHigh,VeryHigh,2,2,Big,Medium,Unacceptable
VeryHigh,VeryHigh,2,2,Big,High,Unacceptable
VeryHigh,VeryHigh,2,4,Small,Low,Unacceptable
VeryHigh,VeryHigh,2,4,Small,Medium,Unacceptable
```

*The file handler f is valid only within the **with** block. It will be closed after it.*

- After executing the above code, if a read is attempted again, an error will be thrown that I/O operation cannot be done on the closed file.

Writing to a File

- Let us review the write operation to a file. In this, first we will read a file that is a famous poem *Life* of *Sarojini Naidu*. The file ***life.txt*** is provided with the worksheet. Then we will select few lines of it and write into a different new file.

```
f = open("C:/Users/BITS-PC/Desktop/life.txt", "r")
lines = f.readlines()
print(lines)
f.close()
```

```
['Life\n', '\n', 'CHILDREN, ye have not lived, to you it seems\n', 'Life is a lovely stalactite of dreams,\n', 'Or ca
rnival of careless joys that leap\n', 'About your hearts like billows on the deep\n', 'In flames of amber and of amet
hyst.\n', '\n', '\n', 'Children, ye have not lived, ye but exist\n', 'Till some resistless hour shall rise and move\n
', 'Your hearts to wake and hunger after love,\n', 'And thirst with passionate longing for the things\n', 'That burn
your brows with blood-red sufferings.\n', '\n', '\n', 'Till ye have battled with great grief and fears,\n', 'And born
e the conflict of dream-shattering years,\n', 'Wounded with fierce desire and worn with strife,\n', 'Children, ye hav
e not lived: for this is life.\n']
```

- This read file in a list is not organized in a way that is easy to read. So ***strip()*** function is used that removes leading and trailing spaces from a string. A string is a sing line of the poem.

```
for line in lines:
    print(line.strip())
```

```
Life

CHILDREN, ye have not lived, to you it seems
Life is a lovely stalactite of dreams,
Or carnival of careless joys that leap
About your hearts like billows on the deep
In flames of amber and of amethyst.


Children, ye have not lived, ye but exist
Till some resistless hour shall rise and move
Your hearts to wake and hunger after love,
And thirst with passionate longing for the things
That burn your brows with blood-red sufferings.


Till ye have battled with great grief and fears,
And borne the conflict of dream-shattering years,
Wounded with fierce desire and worn with strife,
Children, ye have not lived: for this is life.
```

- To make the poem lines more readable the lines can be numbered also as shown below. Please note '\t' is TAB space between the line number and the line.

```
i = 0
for line in lines:
    print(i, '\t', line.strip())
    i = i+1
```

```
0        Life
1
2        CHILDREN, ye have not lived, to you it seems
3        Life is a lovely stalactite of dreams,
4        Or carnival of careless joys that leap
5        About your hearts like billows on the deep
6        In flames of amber and of amethyst.
7
8
9        Children, ye have not lived, ye but exist
10       Till some resistless hour shall rise and move
11       Your hearts to wake and hunger after love,
12       And thirst with passionate longing for the things
13       That burn your brows with blood-red sufferings.
14
15
16       Till ye have battled with great grief and fears,
17       And borne the conflict of dream-shattering years,
18       Wounded with fierce desire and worn with strife,
19       Children, ye have not lived: for this is life.
```

- Note that until now we have not written anything to a file. We have just formatted the print output so that it becomes easier to identify which are the lines we are interested to write into a different file.
- Now let us create a new file in the same location ***favourite.txt*** in which we will write few of our favourite line (e.g. line 3) from the poem.

```
newF = open("C:/Users/BITS-PC/Desktop/favourite.txt", "w")
newF.writelines(lines[3])
newF.close()
```

- Note that, this time the file ***favourite.txt*** is opened in writing mode ("w") because this is the file where we write into.
- After executing the above code we can observe the presence of the new file with the selected line written into. Please note – until the file is closed, the written the content may not appear in the file (e.g. in Google Colab).
- The following code can write multiple lines (3 and 13) through ***for*** iteration.

```
myLines= [3, 13]
with open ("C:/Users/BITS-PC/Desktop/favourite.txt", "w") as newF:
    for line in myLines:
        newF.writelines(lines[line])
```