**Practical No: 14**

**Aim:** File Input/output: Write a program to :

Python Program to Read the Contents of a File

**Course Outcome:** Develop ‘Python’ programs using File Input/outputoperations.

**Requirements: Computer, Python 3.3.34, Vs Code.**

**Theory:**

Python provides inbuilt functions for creating, writing and reading files. There are two types of files that can be handled in python, normal text files and binary files (written in binary language, 0s and 1s).

* **Text files:**In this type of file, Each line of text is terminated with a special character called EOL (End of Line), which is the new line character (‘\n’) in python by default.
* **Binary files:**In this type of file, there is no terminator for a line and the data is stored after converting it into machine understandable binary language.

In this article, we will be focusing on opening, closing, reading, and writing data in a text file.

**File Access Modes**

Access modes govern the type of operations possible in the opened file. It refers to how the file will be used once its opened. These modes also define the location of the **File Handle** in the file. File handle is like a cursor, which defines from where the data has to be read or written in the file. There are 6 access modes in python.

1. **Read Only (‘r’) :**Open text file for reading. The handle is positioned at the beginning of the file. If the file does not exists, raises I/O error. This is also the default mode in which file is opened.
2. **Read and Write (‘r+’) :** Open the file for reading and writing. The handle is positioned at the beginning of the file. Raises I/O error if the file does not exists.
3. **Write Only (‘w’) :** Open the file for writing. For existing file, the data is truncated and over-written. The handle is positioned at the beginning of the file. Creates the file if the file does not exists.
4. **Write and Read (‘w+’)**: Open the file for reading and writing. For existing file, data is truncated and over-written. The handle is positioned at the beginning of the file.
5. **Append Only (‘a’)** : Open the file for writing. The file is created if it does not exist. The handle is positioned at the end of the file. The data being written will be inserted at the end, after the existing data.
6. **Append and Read (‘a+’) :**Open the file for reading and writing. The file is created if it does not exist. The handle is positioned at the end of the file. The data being written will be inserted at the end, after the existing data.

There are three ways to read data from a text file.

read() : Returns the read bytes in form of a string. Reads n bytes, if no n specified, reads the entire file.

File\_object.read([n])

readline() : Reads a line of the file and returns in form of a string.For specified n, reads at most n bytes. However, does not reads more than one line, even if n exceeds the length of the line.

File\_object.readline([n])

readlines() : Reads all the lines and return them as each line a string element in a list.

File\_object.readlines()

**Flowchart:**

**Program:**

a=str(input("Enter the name of the file with .txt extension:"))

file2=open(a,'r')

print(file2.readlines())

file2.close()

**Output/Result:**

Enter the name of the file with .txt extension:Dhiraj.txt

['My self Dhiraj M. Jagtap \n', 'My Id Code is 19CM029']

**Conclusion:**

In this practical we have taken a file name as string a and then opened the file using the open function by giving the a and file access mode parameters. Then printed the whole file using the readlines function and afterwards closed the file using close function. Hence, we have printed or read the contents of the file using file handling functions in python.