Experiment No : 12

Aim : Write python programs to understand different File Handling operations in Python

Description :

Python too supports file handling and allows users to handle files i.e., to read and write files, along with many other file handling options, to operate on files. The concept of file handling has stretched over various other languages, but the implementation is either complicated or lengthy, but like other concepts of Python, this concept here is also easy and short. Python treats file differently as text or binary and this is important. Each line of code includes a sequence of characters and they form text file. Each line of a file is terminated with a special character, called the EOL or End of Line characters like comma {,} or newline character. It ends the current line and tells the interpreter a new one has begun. Let's start with Reading and Writing files.

r: open an existing file for a read operation.

w: open an existing file for a write operation. If the file already contains some data then it will be overridden.

a: open an existing file for append operation. It won't override existing data.

r+: To read and write data into the file. The previous data in the file will not be deleted.

w+: To write and read data. It will override existing data.

a+: To append and read data from the file. It won't override existing data.

Implementation :

Code :

#EXPERIMENT NO 12) Write python programs to understand different File Handling operations in Python

```
# read mode

# open (return file object and catch in a)


a=open("11.txt","r")

print(a.read())

#print(a.read(5))

#print(a.read(10))

#print(a.readline())

#print(a.readline())


a.close()


# append mode.( file handling attach at the end)

# also attach at the end of content, in 11.txt file)

a=open("11.txt","a")
```

```
a.write("file handling")
a.close()
b=open("11.txt","r")
print(b.read())
b.close()

# write (overwrite) .all the previous content will clear.
a=open("11.txt","w")
a.write("file handling")
a.close()
b=open("11.txt","r")
print(b.read())
b.close()

# removing file or delete file. use os module
import os
os.remove("11.txt")
b=open("11.txt","r")
print(b.read())
b.close()
```

```
# creating new empty file no error and no output seen at command prompt

a=open('11.txt',"x")

a.close()

b=open("11.txt","r")

print(b.read())

b.close()


#creating new file . writing and reading in new file


a=open("12.txt","w")

a.write("shridhar")

a.close()

b=open("12.txt","r")

print(b.read())

b.close()
```

Output :

```
HELLO AND WELCOME. PYTHON IS VERY TRENDY.
PYTHON CODE IS EASY TO UNDERSTAND.
HELLO AND WELCOME. PYTHON IS VERY TRENDY.
PYTHON CODE IS EASY TO UNDERSTAND.file handling
file handling
```

Conclusion : Therefore we have successfully implemented python programs to understand different File Handling operations in Python