|  |
| --- |
| Experiment No. 5 |
| Exploring Files and directories: Python program to append data to existing file and then display the entire file |
| Date of Performance:02/02/24 |
| Date of Submission:09/02/24 |

**Experiment No. 5**

**Title:** Exploring Files and directories: Python program to append data to existing file and then display the entire file

**Aim:** To Exploring Files and directories: Python program to append data to existing file and then display the entire file

**Objective:** To Exploring Files and directories

**Theory:**

Directory also sometimes known as a folder are unit organizational structure in computer’s file system for storing and locating files or more folders. Python now supports a number of APIs to list the directory contents. For instance, we can use the Path.iterdir, os.scandir, os.walk, Path.rglob, or os.listdir functions.

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 alike 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.

**Working of open() function**

We use open () function in Python to open a file in read or write mode. As explained above, open ( ) will return a file object. To return a file object we use open() function along with two arguments, that accepts file name and the mode, whether to read or write. So, the syntax being: open(filename, mode). There are three kinds of mode, that Python provides and how files can be opened:

“ r “, for reading.

“ w “, for writing.

“ a “, for appending.

“ r+ “, for both reading and writing

**Code:**

f=open("abc.txt","w")

str=input("Enter the String\n")

f.write(str)

print("String written successfully\n")

f.close()

print("The data in the file is")

f=open("abc.txt","r")

str=f.read()

print(str)

f.close()

print("data read successfully\n")

f=open("abc.txt","w")

print("Enter the messege you want to write")

print("To stop entering messege enter the character @\n")

while str!="@":

str=input()

f.write(str+'\n')

print("Messege write successfully\n")

f.close()

print("The data in the file is")

f=open("abc.txt","r")

str=f.read()

print(str)

print("data read successfully\n")

f.close()

f=open("abc.txt","a+")

print("Enter the messege you want to write")

print("To stop entering messege enter the character @\n")

while str!="@":

str=input()

f.write(str+'\n')

print("Messege write successfully\n")

print("The data in the file is\n")

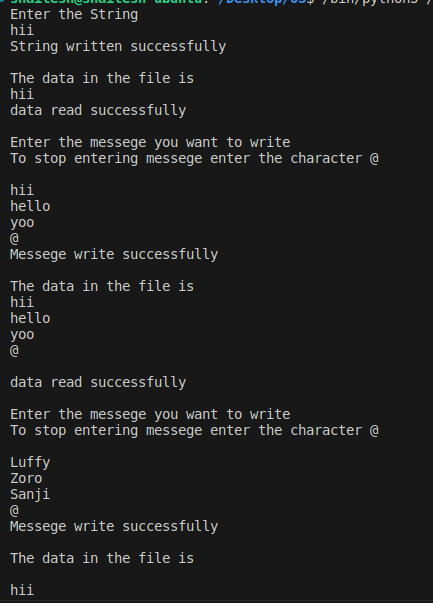
f.seek(0,0)

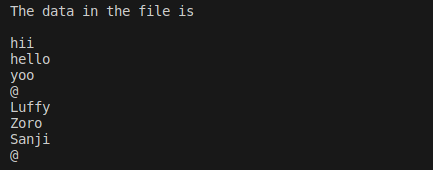
str=f.read()

print(str)

f.close()

**Output:**

****



**Conclusion:** we explored the concepts of working with files and directories in Python, focusing on file handling operations such as appending data to an existing file and displaying the entire file contents.

Directories, also known as folders, serve as organizational units within a computer's file system, providing a structured way to store and locate files and other directories. Python offers various APIs for listing directory contents, including functions like os.listdir(), os.walk(), Path.iterdir(), and more, enabling users to navigate and manipulate directory structures programmatically.

File handling in Python involves operations like reading from and writing to files. Python treats files differently as text or binary, and each line of a text file is terminated with an end-of-line (EOL) character. The open() function is used to open files in different modes, such as reading mode ('r'), writing mode ('w'), and appending mode ('a'). Additionally, the 'r+' mode allows for both reading and writing.