Experiment No. 03

Semester	S.E. Semester IV – Computer Engineering
Subject	Skill Base Lab Course: Python Programming (CSL405)
Subject Professor In-charge	Prof. Swapnil S. Sonawane
Assisting Teachers	Prof. Swapnil S. Sonawane

Student Name	Sejal Dhondkar
Roll Number	19102A0023

Title:

Exploring Files and directories

a. Python program to append data to existing file and then display the entire file.

b. Python program to count number of lines, words and characters in a file.

c. Python program to display file available in current directory.

Objective:

To explore contents of files, directories, and text processing with python

---

Explanation:

Python Files

File handling is an important part of any web application.

Python has several functions for creating, reading, updating, and deleting files.

File Handling

The key function for working with files in Python is the open() function.

The open() function takes two parameters; *filename*, and *mode*.

There are four different methods (modes) for opening a file:

"r" - Read - Default value. Opens a file for reading, error if the file does not exist.

"a" - Append - Opens a file for appending, creates the file if it does not exist.

"w" - Write - Opens a file for writing, creates the file if it does not exist.

"x" - Create - Creates the specified file, returns an error if the file exists.

In addition you can specify if the file should be handled as binary or text mode

"t" - Text - Default value. Text mode

"b" - Binary - Binary mode (e.g. images)

Syntax

To open a file for reading it is enough to specify the name of the file:

f = open("demofile.txt")

The code above is the same as:

f = open("demofile.txt", "rt")

Because "r" for read, and "t" for text are the default values, you do not need to specify them.

Note: Make sure the file exists, or else you will get an error.

Program Code:

a. Python program to append data to existing file and then display the entire file

```
file = input('Enter file name : ')
f1 = open(file, 'a')
print(input('Enter the text to be added : '), file=f1)
f1.close()
f2 = open(file, 'r')
for line in f2:
    print(line.strip())
f2.close()
```

b. Python program to count number of lines, words and characters in a file.

```
file = open(input('Enter file name : '), 'r')
flist = file.readlines()
lines = len(flist)
# print(flist)

words = 0
characters = 0
for line in flist:
    characters += len(line.strip())
    words += len(line.strip().split())

print(lines, words, characters)
```

c. Python program to display file available in current directory

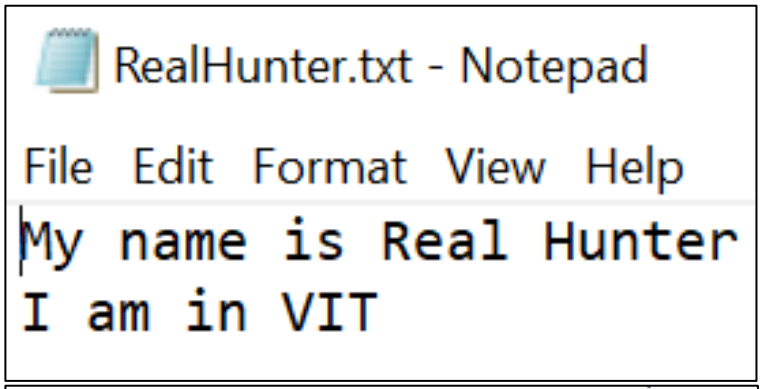
```
import os
a = os.listdir('.')
print(a)
```

Output:

a. Python program to append data to existing file and then display the entire file.

```
Enter file name : RealHunter.txt
Enter the text to be added : My name is Real Hunter
My name is Real Hunter
>>>
========== RESTART: D:\College python\Experimen
Enter file name : RealHunter.txt
Enter the text to be added : I am in VIT
My name is Real Hunter
I am in VIT
>>>
```

b. Python program to count number of lines, words and characters in a file.




```
Enter file name : RealHunter.txt
2 9 33
```

c. Python program to display file available in current directory.

```
['Pract1Consolidated.py', 'Pract1List.py', 'Pract1Pattern.py', 'Pract1Set.py', '
Pract1String.py', 'Pract1Tuples.py', 'Pract2a.py', 'Pract2b.py', 'Pract3a.py', '
Pract3b.py', 'Pract3c.py', 'RealHunter.txt']
```

Conclusion:

Exploration of files and directories is successful by practically implementing programs.
