

Task NO. 8

Date : 10-9-25

IMPLEMENT VARIOUS txt / cs file operations

Aim: To write a Python program for creating and updating student registration details using txt file operations.

algorithm

Step-1 - start

Step-2 - Using open() method, create and write text file "myfile.txt" with student details.

Step-3 - update the new registered student details using append operation in ft.

Step-4 - open the file in read mode and using read() method print the student details.

Step-5 - using seek method print the particular student record.

Step-6 - using tell method print the current position of the file

Step-7 - close the file

Step-8 - stop.

PROGRAM

```
file = open ("student.txt", "w")
input1 = input ("Enter columns names\n")
file.write(input1)
file.write("\n")
n = int(input ("Enter the no. of students"))
for i in range (0,n):
    input2 = input("Enter student's details with for new")
    file.write(input2)
    file.write("\n")
file = open ("student1.txt", "a")
input3 = input ("Enter updated student's details\n")
file.write(input3)
```

output:

Student Details using Read function is;

LITU NO	NAME	AGE	SEX
2305	RAM	20	Male
1920	SHIVA	21	Male
2805	Arjun	19	Male

The length of first line is $10\sqrt{2}$ cm.

15
Output of ReadLine function

Output of Read Line (first student record) function

find the current 2305 RAM 20

~~Position of file pointer.~~

```
file = open("student1.txt", "r")
Print ("Student details using Read function is:")
print(file.read())
Print ("ln")
file.seek(0)
Print ("The length of first line is:")
line = file.readline()
len = len(line)
Print (len)
file.seek(len + 1)
Print ("Output of ReadLine (first student record) function is:")
Print (file.readline())

Print ("In find the current position of file pointer.")
```

```
f = file.tell()
Print(f)
file.close()
```

Result :-

Thus, the python program for creating and updating student registration details using text file operations was executed successfully.

Ex: "Get the best group with
minimum loss giving them the best".
(Name splitting)

(min) taking

(max) taking

"(n - 1) will result in digital sum") taking

() taking

(sum) max = 999

(999) taking

(1 + 999) taking = 1000

of Roberts term will be 1 to 999 taking

as output

Output:- (1000) taking

to defining terms with initial taking

upper case letters :-

lower case letters :- 47 (1000) taking

Digits : 4

(000) taking

5, 47, 4

(000) taking

base pattern of mapping each digit with min
prior which contains prime numbers
and different numbers in sequence

82

Counting cases

construct a Python program whose file name is "merge.txt". To illustrate the below content inside of the file "Python is a high level language developed by Guido van Rossum in 1991" count the total number of uppercase, lowercase and digits used in the text file "merge.txt".

Input

file name: merge.txt

OUTPUT - 5 128, 4

Program to count uppercase, lowercase, and digits in a file.

step-1 : create and write content to the file

with open ("merge.txt", "w") as f:

f.write ("python is a high level language developed by
Guido van Rossum in 1991")

step-2 : open the file for reading.

with open ("merge.txt", "r") as f:

text = f.read()

step 3: Initialize counters

upper - count = 0

lower - count = 0

digit - count = 0

step4 : count uppercase, lowercase and digits

for char in text:

if char.isupper():

upper - count += 1

elif char - is lower():

lower - count + = 1

elif char - is digit (+):

digit - count + = 1

steps . print the result

Print ("uppercase letters": upper - count)

Print ("lowercase letters", lower - count)

Print ("Digits", digit - count)

compact output as required

Print (f" {upper - count}, {lower - count},
{digit - count}).

Output - correct

0.8214 - 0001000

0.2014 - 0000000
0.2014 - 0000000
0.2014 - 0000000



Task -8.3

construct a python program to reveal the above table of student grades from a text file (grades.txt) calculate average grade for each student and print out the result as student name along with their average grade using another txt file (result.txt).

Program

```
# program to read student's grades from a file,  
calculate averages, and save
```

result

```
# Step-1 : Read input data from grades.txt  
with open ("grades.txt", "r") as f:
```

```
    lines = f.read().splitlines()
```

```
# Step-2 : Extract number of students
```

```
n = int (lines[0].strip())
```

```
# Step-3 : Extract weights
```

```
weights = lines[1].strip().split()
```

```
weights = [float(w) for w in weights]
```

```
# Step-4 : Process each student also
```

```
students = []
```

```
for i in range(2, 2+n):
```

```
    pair = lines[i].strip().split()
```

```
    name = pair[0]
```

```
marks = [int(m) for m in pair[1:]])
```

```
total = 0
```

```
for j in range(4):
```

```
    total = total + marks[j] * weights[j]
```

```
students.append([name, round(total, 2)])
```

(Initial 29 - Total 43)

(1 + 39) = 40

(40 * 39) = 1560

(1 + 39) = 40

Repeating step 3

(40 * 19) = 760

(40 * 19) = 760

(40 * 19) = 760

Repeating step 3

(40 * 19) = 760

(40 * 19) = 760

Output

Gaurav \rightarrow 169.0

Abhinav \rightarrow 138.0

Harvard \rightarrow 152.0

Jai \rightarrow 163.0

Kavi \rightarrow 188.0

Steps : write results into result.txt

with open ("result.txt", "w") as f:

for name, avg in students:

f.write (name + " → " + str(avg) + "\n")

print ("Average grades have been written to result.txt")

VEL TECH	
EX NO.	8
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	3
VIVA VOCE (5)	5
RECORD (5)	
TOTAL (20)	
SIGN WITH DATE	15—

Result:- The programs using txt/CSV file operations
are executed & verified successfully.