FILE HANDLING

Files are used to store data permanently and can be retrieved later.

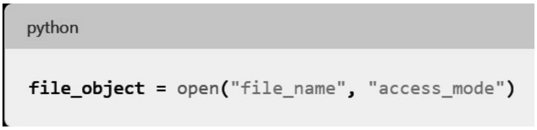
Type of Files

1. Text Files
2. Binary Files
3. CSV Files

Steps for File Handling:

1. Open File
2. Processing file i.e Read or Write from/to file
3. Close File

1. Opening Files:



python

```python
file_object = open("file_name", "access_mode")
```

Let's break it down:

I. **file_object:** This is a variable that will hold the file object returned by the `open()` function. You'll use this variable to perform operations like reading from or writing to the file.

II. **open():** This is the built-in function used to open files in Python.

III. **access_mode:** This specifies the mode in which you want to open the file.

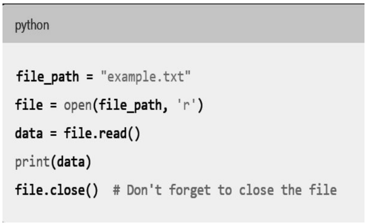
There are two ways of opening a file in Python

a. open()

b. Open using ‘with’ statement.

open() Function:	**with Statement:**
● It's like turning the key to open a door. ● You use it to open a file and get access to its contents.	● It's like having a helper who opens and closes the door for you automatically. ● You tell the helper what you want to do with the door (read, write,

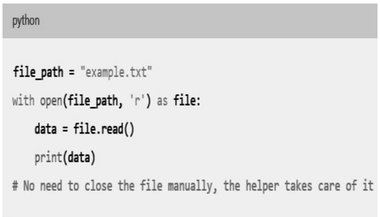
- After you're done reading or writing, you have to remember to close the file yourself, just like locking the door when you're done.



```python
file_path = "example.txt"
file = open(file_path, 'r')
data = file.read()
print(data)
file.close()  # Don't forget to close the file
```

etc.), and they handle everything for you.

- They make sure the door is properly closed even if something unexpected happens, like a sudden gust of wind.



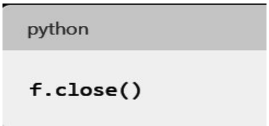
```python
file_path = "example.txt"
with open(file_path, 'r') as file:
    data = file.read()
    print(data)
# No need to close the file manually, the helper takes care of it
```

Best Choice:

- Using `with` is usually better because it's safer and saves you from having to remember to close the file.
- It's like having a reliable assistant who ensures everything is handled smoothly.

2. Closing Files:

- It's important to close files after you're done working with them to release system resources.

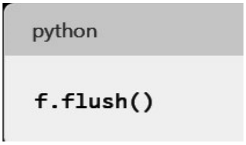


```python
f.close()
```

- Not closing files can lead to resource leaks and other issues.

3. Buffering and Flushing:

- When writing to a file, data might be stored in an output buffer until the file is closed or explicitly flushed.
- The `flush()` method forces the data waiting in the buffer to be immediately written into the file without waiting for the file to be closed.



```python
f.flush()
```

Access Specifiers in Files:

Access Mode for Text Files	Access Mode for Binary Files	Access Mode for CSV Files	Description	File Pointer Position
r	rb	r	Read mode. Opens a file for reading. If the file does not exist, open() raises a FileNotFoundError.	Beginning of File
r+	rb+		It opens the file for both reading and writing. If the file does not exist, open() raises a FileNotFoundError.	Beginning of File
w	wb	w	Write mode. It opens the file for writing only. If the file exists, the content of the file will be removed. If the file does not exist, it is created.	Beginning of File
w+	wb+		The w+ mode opens the file for both writing and reading. Like w, if the file exists, then the content of the file will be removed. If the file does not exist, it is created.	Beginning of File
a+	ab+		The a+ mode opens the file for both appending and reading. In this, the new content is added after the existing content. If the file does not exist, it is created.	End of File
a	ab	a	The a mode opens the file for appending. In this, the new content is added after the existing content. If the file does not exist, it creates a new file.	End of File

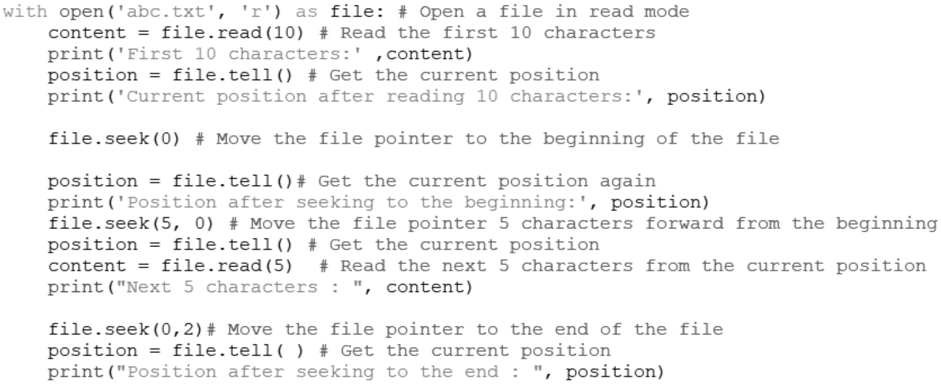
The default mode for file opening is “r” read mode. If we didn’t specify mode during the opening of the file then it will automatically open the file in read mode.

File Object Methods (seek( ) & tell( ))

Method	Prototype	Description
seek( )	**Syntax:** **<file**_object>.seek(<offset>,<whence>) offset: Number of bytes to move the file pointer. whence: Position reference: ● 0: Beginning of the file (default). ● 1: Current file position. ● 2: End of the file.	● It is used to change the position of the **File Handle** to a given specific position. ● A file handle is like a cursor, which defines from where the data has to be read or written in the file. **Note:** In the case of a text file we can use only ‘**0’** as a reference point.
tell( )	**Syntax: <file_object>.tell( )** i.e. position=f.tell( ) where f is the file handle.	● returns the current position of the file object. ● It takes no parameters ● It returns an integer value.

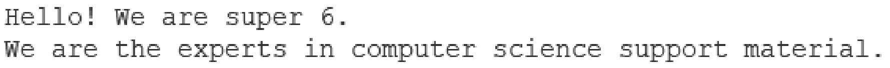
Example:

Python Code:



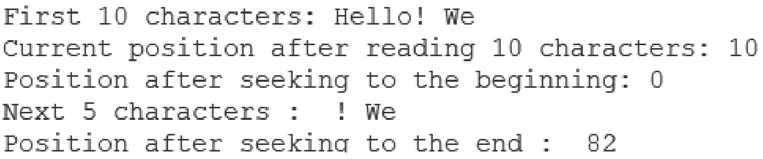
```
with open('abc.txt', 'r') as file: # Open a file in read mode
    content = file.read(10) # Read the first 10 characters
    print('First 10 characters:' ,content)
    position = file.tell() # Get the current position
    print('Current position after reading 10 characters:', position)

    file.seek(0) # Move the file pointer to the beginning of the file

    position = file.tell()# Get the current position again
    print('Position after seeking to the beginning:', position)
    file.seek(5, 0) # Move the file pointer 5 characters forward from the beginning
    position = file.tell() # Get the current position
    content = file.read(5)  # Read the next 5 characters from the current position
    print("Next 5 characters : ", content)

    file.seek(0,2)# Move the file pointer to the end of the file
    position = file.tell( ) # Get the current position
    print("Position after seeking to the end : ", position)
```

Content of Text File:



```
Hello! We are super 6.
We are the experts in computer science support material.
```

Output after executing above code:



```
First 10 characters: Hello! We
Current position after reading 10 characters: 10
Position after seeking to the beginning: 0
Next 5 characters :  ! We
Position after seeking to the end :  82
```

Text Files:

- It stores information in the form of ASCII or Unicode characters
- 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.
- The file extension will be .txt

Working with Text Files:

1. Reading data from a file.
2. Writing data into a file.

Reading data from files

There are three ways to read data from text files:

1. read function i.e. read( )
2. readline function i.e. readline( )
3. readlines function i.e. readlines( )

read( ): It is used in text files to read a specified number of data bytes from the file. It returns the result in the form of a string.

Syntax: file_object.read( )

file_pointer.read(n): It will read the maximum number of n bytes/characters from the file.

f.read(7) # It will read 7 bytes/characters from the position of the file pointer.

file_pointer.read( ): It will read the entire content of the file.

f.read( ) # It will read all the data of the file from the position of the file pointer.

readline( ): It will read one complete line in one go from the file. It returns the data in the form of a string.

Syntax: file_object.readline( )

file_pointer.readline( ): It will read the entire line.

f.readline( ) #it will read one complete line in one go.

file_pointer.readline(n): It will read the first ‘n’ bytes from the file.

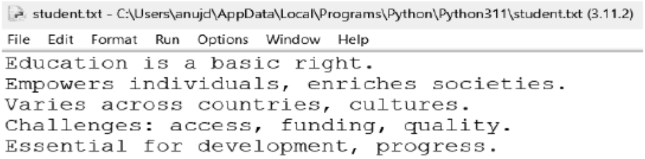
f.readline(5) #it will read the first 5 characters/bytes from the file.

readlines( ): It will return all the lines of the file as the elements of the list. I.e. the 1st line of the file will be the first element of the list and so on.

Syntax: file_object.readlines( )

file_pointer.readlines( ): It will read all the lines of the file as the elements of the list.

File Content

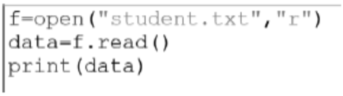


student.txt - C:\Users\anujd\AppData\Local\Programs\Python\Python311\student.txt (3.11.2)

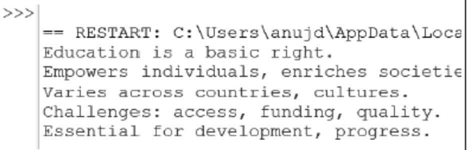
```
Education is a basic right.
Empowers individuals, enriches societies.
Varies across countries, cultures.
Challenges: access, funding, quality.
Essential for development, progress.
```

Code

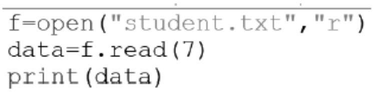
Output



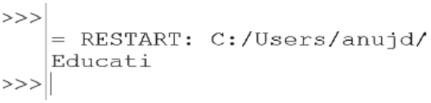
```
f=open("student.txt","r")
data=f.read()
print(data)
```



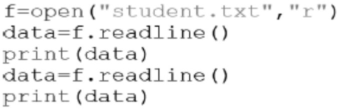
```
>>>
== RESTART: C:\Users\anujd\AppData\Loca
Education is a basic right.
Empowers individuals, enriches societie
Varies across countries, cultures.
Challenges: access, funding, quality.
Essential for development, progress.
```



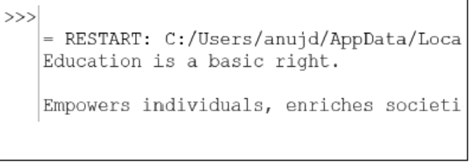
```
f=open("student.txt","r")
data=f.read(7)
print(data)
```



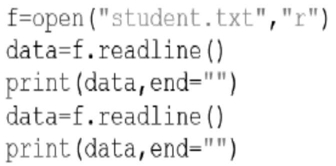
```
>>>
= RESTART: C:/Users/anujd/
Educati
>>>
```



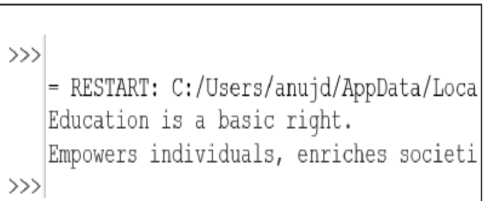
```
f=open("student.txt","r")
data=f.readline()
print(data)
data=f.readline()
print(data)
```



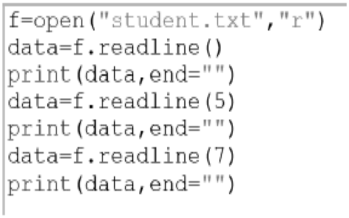
```
>>>
= RESTART: C:/Users/anujd/AppData/Loca
Education is a basic right.

Empowers individuals, enriches societi
```



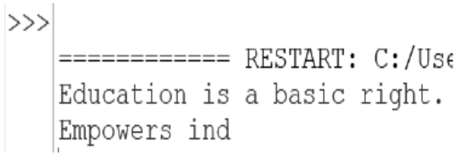
```
f=open("student.txt","r")
data=f.readline()
print(data,end="")
data=f.readline()
print(data,end="")
```



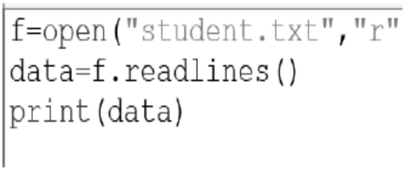
```
>>>
    = RESTART: C:/Users/anujd/AppData/Loca
    Education is a basic right.
    Empowers individuals, enriches societi
>>>
```



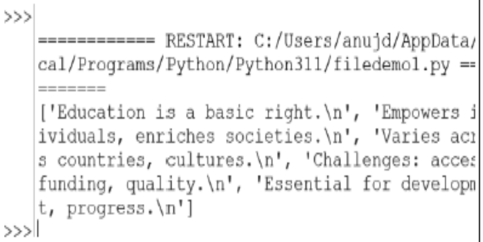
```
f=open("student.txt","r")
data=f.readline()
print(data,end="")
data=f.readline(5)
print(data,end="")
data=f.readline(7)
print(data,end="")
```



```
>>>
    ============ RESTART: C:/Use
    Education is a basic right.
    Empowers ind
```

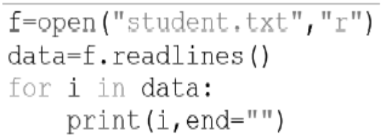


```
f=open("student.txt","r"
data=f.readlines()
print(data)
```

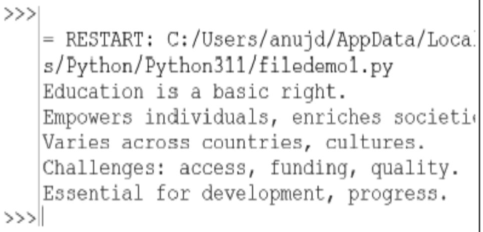


```
>>>
    ============ RESTART: C:/Users/anujd/AppData/
    cal/Programs/Python/Python311/filedemo1.py ==
    =======
    ['Education is a basic right.\n', 'Empowers i
    ividuals, enriches societies.\n', 'Varies acr
    s countries, cultures.\n', 'Challenges: acces
    funding, quality.\n', 'Essential for developm
    t, progress.\n']
>>>
```

Result is in the form of a list.



```
f=open("student.txt","r")
data=f.readlines()
for i in data:
    print(i,end="")
```

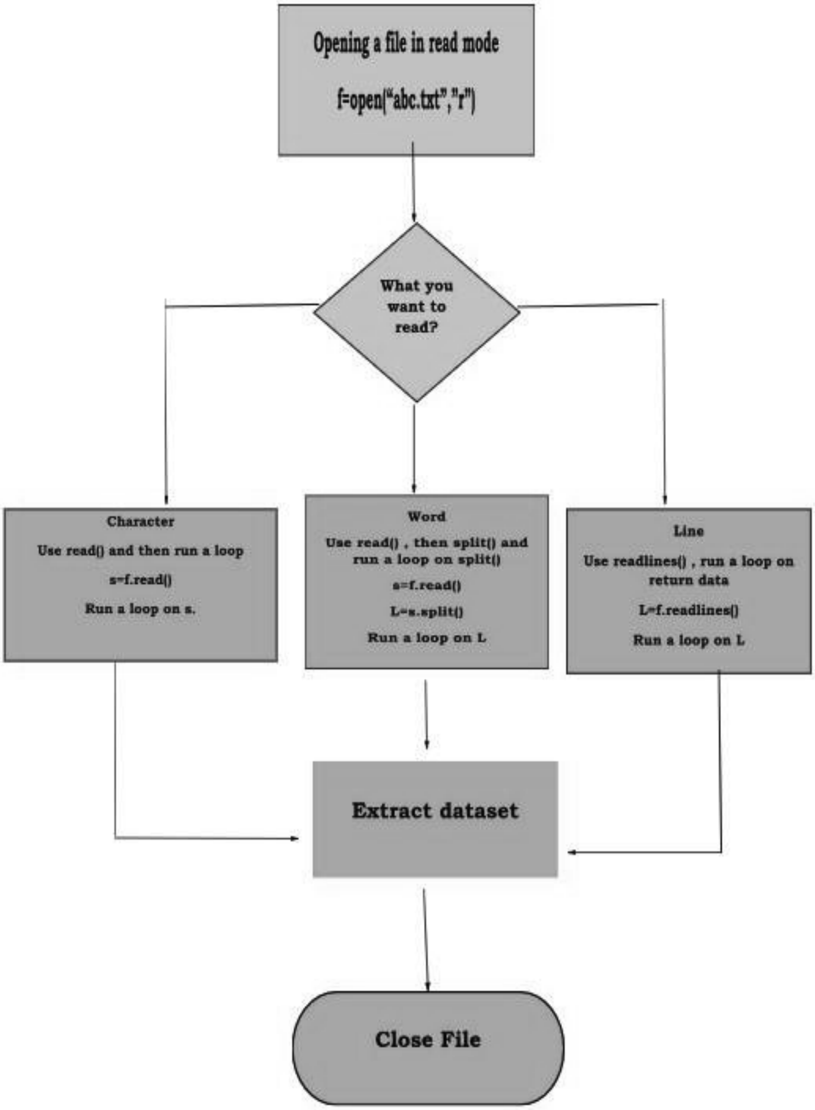


```
>>>
    = RESTART: C:/Users/anujd/AppData/Loca
    s/Python/Python311/filedemo1.py
    Education is a basic right.
    Empowers individuals, enriches societi
    Varies across countries, cultures.
    Challenges: access, funding, quality.
    Essential for development, progress.
>>>
```

Tips on writing text file code in exams:

- Let the file name be “abc.txt”.
- Read the question carefully and find out what has to be read from the file.

Follow the below flow chart to write the code.



Opening a file in read mode

f=open(“abc.txt”,"r")

What you want to read?

Character

Use read() and then run a loop

s=f.read()

Run a loop on s.

Word

Use read() , then split() and run a loop on split()

s=f.read()

L=s.split()

Run a loop on L

Line

Use readlines() , run a loop on return data

L=f.readlines()

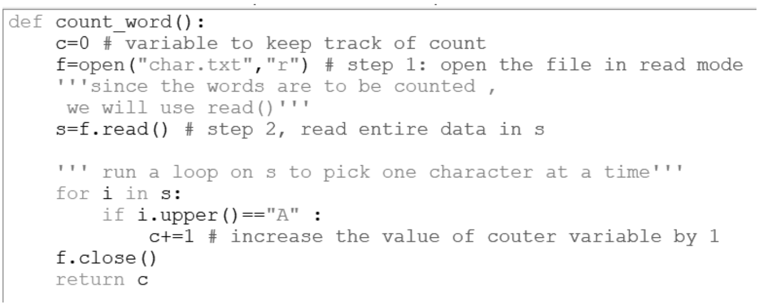
Run a loop on L

Extract dataset

Close File

Example 1:

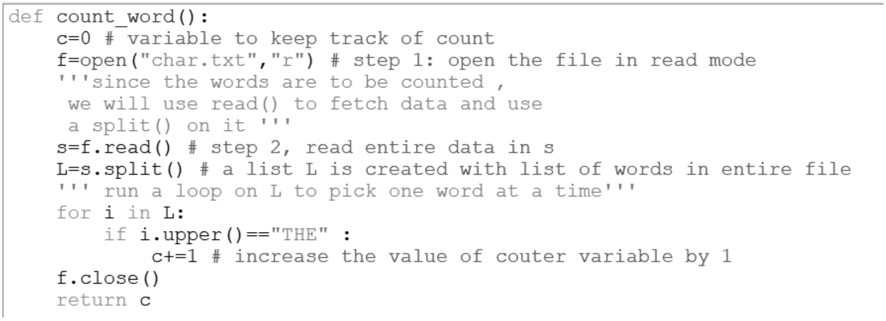
Write a function count_char() that reads a file named "char.txt" counts the number of times character "a" or "A" appears in it.



```python
def count_word():
    c=0 # variable to keep track of count
    f=open("char.txt","r") # step 1: open the file in read mode
    '''since the words are to be counted ,
     we will use read()'''
    s=f.read() # step 2, read entire data in s

    ''' run a loop on s to pick one character at a time'''
    for i in s:
        if i.upper()=="A" :
            c+=1 # increase the value of couter variable by 1
    f.close()
    return c
```

Example 2:

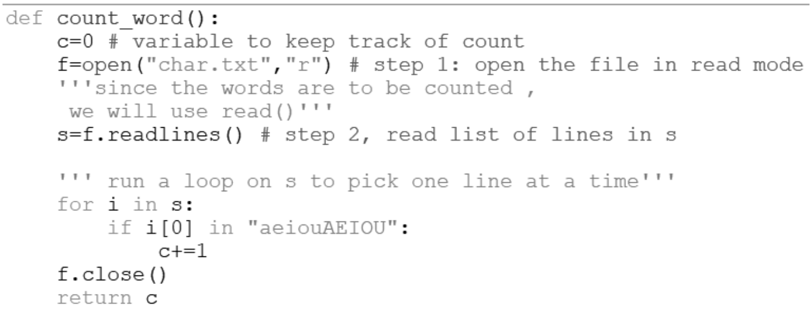
Write a function count_word() that reads a text file named "char.txt" and returns the number of times word " the" exists.



```python
def count_word():
    c=0 # variable to keep track of count
    f=open("char.txt","r") # step 1: open the file in read mode
    '''since the words are to be counted ,
     we will use read() to fetch data and use
     a split() on it '''
    s=f.read() # step 2, read entire data in s
    L=s.split() # a list L is created with list of words in entire file
    ''' run a loop on L to pick one word at a time'''
    for i in L:
        if i.upper()=="THE" :
            c+=1 # increase the value of couter variable by 1
    f.close()
    return c
```

Example 3:

Write a function count_line() that reads a text file named "char.txt" and returns the number of lines that start with a vowel.



```python
def count_word():
    c=0 # variable to keep track of count
    f=open("char.txt","r") # step 1: open the file in read mode
    '''since the words are to be counted ,
     we will use read()'''
    s=f.readlines() # step 2, read list of lines in s

    ''' run a loop on s to pick one line at a time'''
    for i in s:
        if i[0] in "aeiouAEIOU":
            c+=1
    f.close()
    return c
```

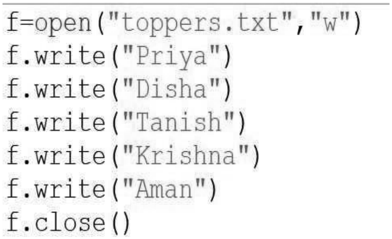
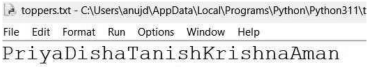
Writing data into Files

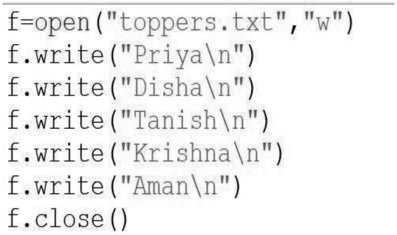
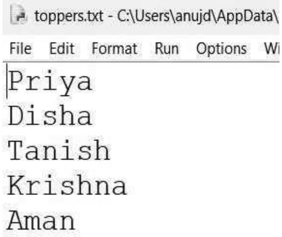
1. **write( ):** It takes a string as an input and writes it into the file.
    a. **Syntax**: file_object.write(string)
    b. i.e. f.write(“Hello World”)
2. **writelines( ):** It is used to write multiple lines as a list of strings into the file. In this, each element of the list will be treated as a separate line in the file.
    a. **Syntax:** file_object.writelines(list of strings)

i.e. data=[“I am a student of DOE”, “I studies in class 12th”]

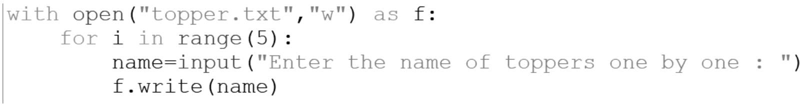
>>>f.writelines(data)

Code	**Output**


`f=open("toppers.txt","w")`
`f.write("Priya")`
`f.write("Disha")`
`f.write("Tanish")`
`f.write("Krishna")`
`f.write("Aman")`
`f.close()`
	Content in “Topper.txt” file
	
	 toppers.txt - C:\Users\anujd\AppData\Local\Programs\Python\Python311\t
	File Edit Format Run Options Window Help
	`PriyaDishaTanishKrishnaAman`
	But, we want these names in separate lines.

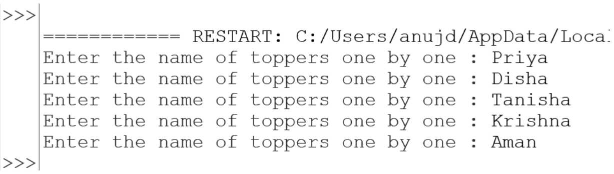

`f=open("toppers.txt","w")`
`f.write("Priya\n")`
`f.write("Disha\n")`
`f.write("Tanish\n")`
`f.write("Krishna\n")`
`f.write("Aman\n")`
`f.close()`
	
	 toppers.txt - C:\Users\anujd\AppData\
	File Edit Format Run Options Wi
	`Priya`
	`Disha`
	`Tanish`
	`Krishna`
	`Aman`

Code:



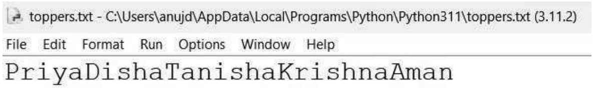
```
with open("topper.txt","w") as f:
    for i in range(5):
        name=input("Enter the name of toppers one by one : ")
        f.write(name)
```

Output:



```
>>>
    ============ RESTART: C:/Users/anujd/AppData/Loca
    Enter the name of toppers one by one : Priya
    Enter the name of toppers one by one : Disha
    Enter the name of toppers one by one : Tanisha
    Enter the name of toppers one by one : Krishna
    Enter the name of toppers one by one : Aman
>>>
```

Content of "Topper.txt" File:



toppers.txt - C:\Users\anujd\AppData\Local\Programs\Python\Python311\toppers.txt (3.11.2)

```
PriyaDishaTanishaKrishnaAman
```

Question: Write a program in python with reference to above program, the content of the textfiles should be in different lines.

I.e.

Priya

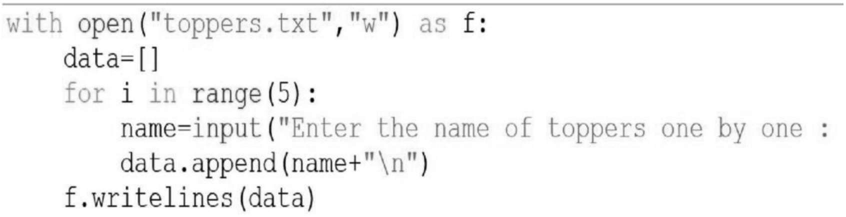
Disha

Tanisha

Krishna

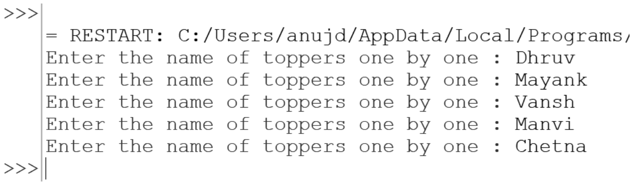
Aman

Code



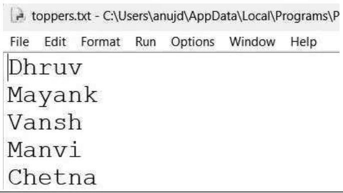
```
with open("toppers.txt","w") as f:
    data=[]
    for i in range(5):
        name=input("Enter the name of toppers one by one :
        data.append(name+"\n")
    f.writelines(data)
```

Output



```
>>>
    = RESTART: C:/Users/anujd/AppData/Local/Programs/
    Enter the name of toppers one by one : Dhruv
    Enter the name of toppers one by one : Mayank
    Enter the name of toppers one by one : Vansh
    Enter the name of toppers one by one : Manvi
    Enter the name of toppers one by one : Chetna
>>>
```

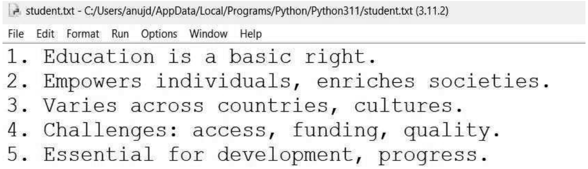
Content of “**Toppers.txt**” file:


toppers.txt - C:\Users\anujd\AppData\Local\Programs\P
File Edit Format Run Options Window Help

```
Dhruv
Mayank
Vansh
Manvi
Chetna
```

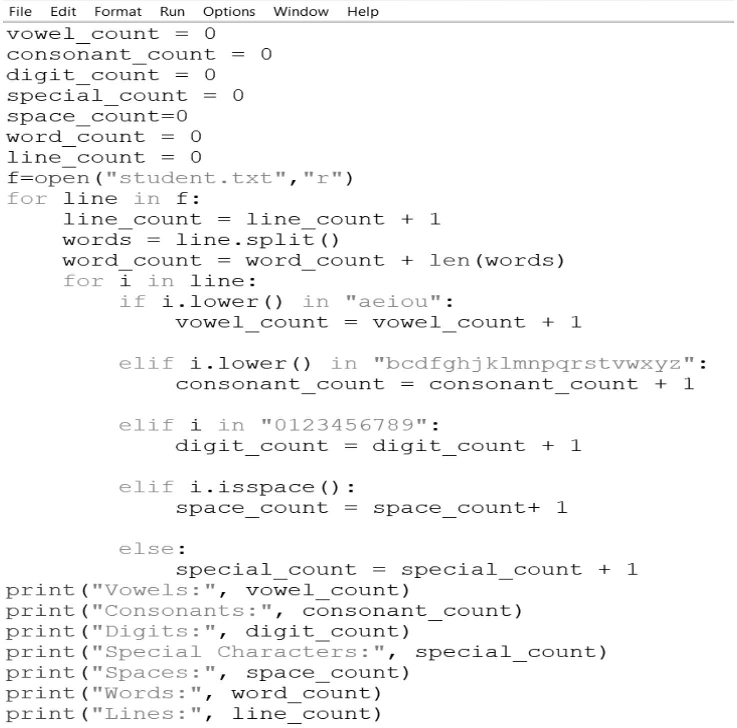
Write a program in python to count vowels, consonants, digits, spaces, special characters, spaces, words and lines from a text file named “student.txt”.

Content of File:


student.txt - C:/Users/anujd/AppData/Local/Programs/Python/Python311/student.txt (3.11.2)
File Edit Format Run Options Window Help

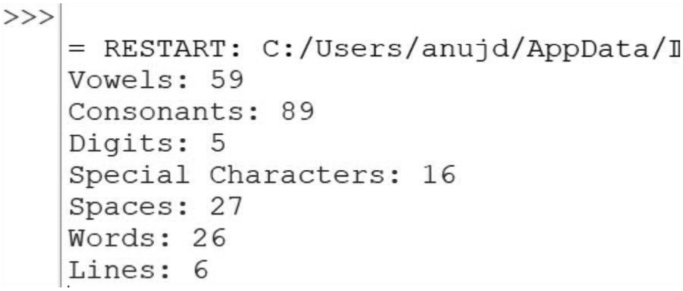
```
1. Education is a basic right.
2. Empowers individuals, enriches societies.
3. Varies across countries, cultures.
4. Challenges: access, funding, quality.
5. Essential for development, progress.
```

Code



```python
vowel_count = 0
consonant_count = 0
digit_count = 0
special_count = 0
space_count=0
word_count = 0
line_count = 0
f=open("student.txt","r")
for line in f:
    line_count = line_count + 1
    words = line.split()
    word_count = word_count + len(words)
    for i in line:
        if i.lower() in "aeiou":
            vowel_count = vowel_count + 1

        elif i.lower() in "bcdfghjklmnpqrstvwxyz":
            consonant_count = consonant_count + 1

        elif i in "0123456789":
            digit_count = digit_count + 1

        elif i.isspace():
            space_count = space_count+ 1

        else:
            special_count = special_count + 1
print("Vowels:", vowel_count)
print("Consonants:", consonant_count)
print("Digits:", digit_count)
print("Special Characters:", special_count)
print("Spaces:", space_count)
print("Words:", word_count)
print("Lines:", line_count)
```

Output:



```
>>>
    = RESTART: C:/Users/anujd/AppData/I
    Vowels: 59
    Consonants: 89
    Digits: 5
    Special Characters: 16
    Spaces: 27
    Words: 26
    Lines: 6
```

Lab Exercise:

1. Define a function SGcounter() that counts and displays the number of S and G present in a text file ‘A.txt”

   e.g., SAGAR JOON IS GOING TO MARKET.

   It will display S:2 G:2

2. Write a function in Python that counts the number of “is”, “am” or “are” words present in a text file “HELLO.TXT”. If the “HELLO.TXT” contents are as follows: *Here are two sentences that contain "is," "am," or "are":*

   “She is studying for her final exams.
   We are planning a trip to the mountains next weekend.”
   The output of the function should be: Count of is/am/are in file: 2

3. Write a method in Python to read lines from a text file HELLO.TXT to find and display the occurrence of the word “hello”.

4. Write a user-defined function named Count() that will read the contents of a text file named “India.txt” and count the number of lines which start with either “I” or “T”.

   E.g. In the following paragraph, 2 lines are starting with “I” or “T”:
   “*The Indian economy is one of the largest and fastest-growing in the world, characterized by a diverse range of industries including agriculture, manufacturing, services, and information technology. It boasts a sizable consumer base and a dynamic entrepreneurial spirit. However, it also faces challenges such as income inequality, poverty, and infrastructure gaps, which the government continually addresses through policy reforms and initiatives to foster sustainable growth .”*

5. Write a method in Python to read lines from a text file AZAD.TXT and display those lines, which start with the alphabet ‘T’.