

Data File Handling *in Python*

**Excerpts from
Classroom and Online Classes**

**Coded by
Somnath Paul Choudhury**

ispcin@gmail.com

It's also relevant for the Engineering Graduates

So you have started Python for the Semester and have already mastered the basics.

Data can be stored in secondary storage so that it can be used time and again.

If we want to store something we need to open a file first then perform write operation finally we save the file and close the file. Sometimes we may just open the file to read it or add something more to it.

In Python we deal with Text and Binary files.

Text files stored in ASCII format or UNICODE format have End Of Line delimiters. Binary has no such property and everything is directly stored in machine readable format and size is also less.

Working with text files

Task We intend to create a txt file in the working directory

```
# importing os module
```

```
>>> import os
```

```
# calling the function getcwd() to check the current working directory  
# the working directory is displayed in the next line when we press Enter key
```

```
>>> os.getcwd()  
'C:\\Program Files\\PythonCodes'
```

```
# Working directory is located in 'C:\\Program Files\\PythonCodes'
```

```
>>> file=open("spc000.txt","x")  
Traceback (most recent call last):  
  File "<pyshell#4>", line 1, in <module>  
    file=open("spc000.txt","x")  
PermissionError: [Errno 13] Permission denied: 'spc000.txt' #Program Files is a  
forbidden area
```

So we have to change the working directory as shown below

```
>>> os.chdir(r'C:\\Users\\Som\\AppData\\Local\\Programs\\Python\\Python38')  
>>> os.getcwd()  
'C:\\Users\\Som\\AppData\\Local\\Programs\\Python\\Python38'
```

```
# Changed it!!!
```

```
>>> file=open("spc000.txt","x")
```

```
>>> file.write("Hello")
```

```
>>> file.close()
```

```
>>> #File is created
```

Name	Date modified	Type	Size
spc000	07-02-2021 12:04	Text Document	1 KB
spc	01-02-2021 08:49	DAT File	1 KB
spc02notes	30-01-2021 08:41	Text Document	0 KB
spc01notes	30-01-2021 08:40	Text Document	1 KB

Let's try to create a multiline text disk file

Place the lines as string elements in a list

```
>>> mylist=["This is the First Line","\nThis is the Second Line",
"\nThis is the Third Line"]
```

Prepare a file for writing

```
>>> file=open("spc000.txt","w")
>>> file.writelines(mylist)
>>> file.close()
```

Now let's read the file

```
>>> file=open("spc000.txt","r")
>>> print(file.read())
```

```
This is the First Line
This is the Second Line
This is the Third Line
```

The contents are also displayed as shown above

Let's try to move around the file using functions seek() and tell()

```
>>> file.tell()
71
>>>
```

tell() returns the current position of the file also referred to as the file pointer

seek() can change the current position of the file pointer

```
>>> file.seek(0)
0
>>> file.seek(49)
49
>>> print(file.read())
This is the Third Line
>>>
```

Here we can see seek(49) moves the file pointer to the beginning of the third line

Let's read the text file once again and segregate the upper case , lower case , vowels and consonants in separate lists.

```
>>>

>>> file.close()
>>> file=open("spc000.txt","r")
>>> mylist=[]
>>> for i in file.read():
    if(i != '\n'):
        if( i != ' '):
            mylist.append(i)

>>> mylist
['T', 'h', 'i', 's', 'i', 's', 't', 'h', 'e', 'F', 'i', 'r', 's', 't', 'L', 'i', 'n', 'e', 'T', 'h', 'i', 's', 'i', 's', 't', 'h', 'e', 'S',
'e', 'c', 'o', 'n', 'd', 'L', 'i', 'n', 'e', 'T', 'h', 'i', 's', 'i', 's', 't', 'h', 'e', 'T', 'h', 'i', 'r', 'd', 'L', 'i', 'n', 'e']
>>>
```

Now let us segregate

```
>>> upplist=[]
>>> lowlist=[]
>>> conslist=[]
>>> vowlist=[]
>>> for i in mylist:
    if(i.isupper()==True):
        upplist.append(i)
```

For Limited Circulation Only , authored by Somnath PaulChoudhury

```
>>> upplist
['T', 'F', 'L', 'T', 'S', 'L', 'T', 'T', 'L']
```

```
>>> for i in mylist:
    if(i.islower()==True):
        lowlist.append(i)
```

```
>>> lowlist
['h', 'i', 's', 'i', 's', 't', 'h', 'e', 'i', 'r', 's', 't', 'i', 'n', 'e', 'h', 'i', 's', 'i', 's', 't', 'h', 'e', 'e', 'c', 'o', 'n', 'd',
'i', 'n', 'e', 'h', 'i', 's', 'i', 's', 't', 'h', 'e', 'h', 'i', 'r', 'd', 'i', 'n', 'e']
```

```
>>> for i in mylist:
    if(i in 'aeiouAEIOU'):
        vowlist.append(i)
```

```
>>> vowlist
['i', 'i', 'e', 'i', 'i', 'e', 'i', 'i', 'e', 'e', 'o', 'i', 'e', 'i', 'i', 'e', 'i', 'i', 'e']
```

```
>>> for i in mylist:
    if(i not in 'aeiouAEIOU'):
        conslist.append(i)
```

```
>>> conslist
['T', 'h', 's', 's', 't', 'h', 'F', 'r', 's', 't', 'L', 'n', 'T', 'h', 's', 's', 't', 'h', 'S', 'c', 'n', 'd', 'L', 'n', 'T', 'h',
's', 's', 't', 'h', 'T', 'h', 'r', 'd', 'L', 'n']
>>>
```

We can count the lines , words and characters from a disk file

```
import os
spcfile=input("Enter a filename ")
lc=0
wc=0
cc=0
file=open(spcfile,"r")
print(file.read())
```

with open(spcfile, 'r') as f:

```
    for line in f:
        lc += 1
        words=line.split()
        wc+=len(words)
        cc=cc+len(line)
print("Number of lines : ",lc," Number of words : ",wc," Number of characters : ",cc)
```

Enter a filename spcnotes.txt

Hello How are you

Hi How are you

Hello Hello How are you

Hallo, Guten Morgen

Number of lines : 4 Number of words : 16 Number of characters : 76

Understanding the function open()

In the previous section we have seen the function open() that can open a file. The name of the file it will open is the first argument. The second argument is the mode in which it will open the file. When we use the second argument as x the file if not exists already will be created. We can use the mode as w , r or a for writing and erasing any previous contents, reading and appending.

The write() and read() methods we have used with txt files use string parameters. While working with the binary file conversion of data while reading and writing is required. Python has a module pickle for the purpose. Pickling is converting a data structure into byte stream for writing. Unpickling is the reverse, conversion of byte stream into original structure before reading.

The following code demonstrates how to dump in a binary file 100 random numbers which is added to a list and read back the same

```
>>> import os

>>> os.getcwd()

'C:\\Users\\Som\\AppData\\Local\\Programs\\Python\\Python38'

>>> import random

>>> mylist=[]

>>> for i in range(100):

    mylist.append(random.randint(3,300))

>>> mylist

[60, 190, 292, 178, 280, 197, 263, 21, 18, 183, 112, 40, 221, 144, 243, 27, 244, 187,
290, 8, 43, 143, 163, 31, 176, 223, 113, 45, 228, 176, 25, 246, 32, 294, 286, 209, 296,
55, 234, 193, 163, 266, 80, 256, 262, 30, 226, 246, 246, 73, 214, 196, 225, 144, 83,
165, 150, 55, 25, 124, 133, 152, 95, 280, 92, 145, 102, 235, 94, 40, 122, 197, 188, 213,
56, 154, 180, 256, 108, 166, 277, 86, 254, 255, 98, 136, 101, 289, 64, 162, 260, 159,
280, 177, 213, 11, 35, 284, 182, 88]

>>> import pickle

>>> spcfile=open("spcnew00.dat","wb")

>>> mylist.sort()

>>> mylist
```

For Limited Circulation Only , authored by Somnath PaulChoudhury

```
[8, 11, 18, 21, 25, 25, 27, 30, 31, 32, 35, 40, 40, 43, 45, 55, 55, 56, 60, 64, 73, 80, 83,
86, 88, 92, 94, 95, 98, 101, 102, 108, 112, 113, 122, 124, 133, 136, 143, 144, 144, 145,
150, 152, 154, 159, 162, 163, 163, 165, 166, 176, 176, 177, 178, 180, 182, 183, 187,
188, 190, 193, 196, 197, 197, 209, 213, 213, 214, 221, 223, 225, 226, 228, 234, 235,
243, 244, 246, 246, 246, 254, 255, 256, 256, 260, 262, 263, 266, 277, 280, 280, 280,
284, 286, 289, 290, 292, 294, 296]
```

```
>>> pickle.dump(mylist,spcfile)
```

```
>>> spcfile.close()
```

```
>>> spcfile=open("spcnew00.dat","rb")
```

```
>>> try:
```

```
    while(True):
```

```
        i=pickle.load(spcfile)
```

```
        print(i, end="")
```

```
except EOFError:
```

```
    pass
```

```
[8, 11, 18, 21, 25, 25, 27, 30, 31, 32, 35, 40, 40, 43, 45, 55, 55, 56, 60, 64, 73, 80, 83,
86, 88, 92, 94, 95, 98, 101, 102, 108, 112, 113, 122, 124, 133, 136, 143, 144, 144, 145,
150, 152, 154, 159, 162, 163, 163, 165, 166, 176, 176, 177, 178, 180, 182, 183, 187,
188, 190, 193, 196, 197, 197, 209, 213, 213, 214, 221, 223, 225, 226, 228, 234, 235,
243, 244, 246, 246, 246, 254, 255, 256, 256, 260, 262, 263, 266, 277, 280, 280, 280,
284, 286, 289, 290, 292, 294, 296]
```

```
>>> spcfile.close()
```

```
>>>
```

If we want to display only numbers divisible by 5 , we modify it as

```
>>> spcfile.close()
```

```
>>> spcfile=open("spcnew00.dat","rb")
```

```
>>> try:
```

For Limited Circulation Only , authored by Somnath PaulChoudhury

```

while(True):
    i=pickle.load(spcfile)
    for j in i:
        if(j%5==0):
            print(j, end=" , ")
except EOFError:
    pass

25 , 25 , 30 , 35 , 40 , 40 , 45 , 55 , 55 , 60 , 80 , 95 , 145 , 150 , 165 ,
180 , 190 , 225 , 235 , 255 , 260 , 280 , 280 , 280 , 290 ,

>>>

```

This code demonstrates simple dumping of numbers in a .dat file and displaying it

```

>>> import os
>>> os.getcwd()
'C:\\Users\\Som\\AppData\\Local\\Programs\\Python\\Python38'
>>> import pickle
>>> spcfile=open("spc2120.dat","wb")
>>> while True:
    x=int(input("Enter a number "))
    pickle.dump(x,spcfile)
    response=input("Continue? ")
    if(response=="n" or response=="N"):

```

For Limited Circulation Only , authored by Somnath PaulChoudhury

break

Enter a number 21

Continue? y

Enter a number 84

Continue? y

Enter a number 79

Continue? y

Enter a number 100

Continue? y

Enter a number -99

Continue? y

Enter a number 1

Continue? n

```
>>> spcfile=open("spc2120.dat","rb")
```

```
>>> try:
```

```
    while True:
```

```
        i=pickle.load(spcfile)
```

```
        print(i)
```

```
except EOFError:
```

```
    pass
```

21

84

79

100

-99

1

```
>>> spcfile.close()
```

```
>>>
```

Pickling unpickling class objects

```
>>> import pickle
```

```
>>> class item:
```

```
    def __init__(self, name,qty):
```

```
        self.nm=name
```

```
        self.qy=qty
```

```
    def showme(self):
```

```
        print(self.nm, " " ,self.qy)
```

```
>>> with open("spnotes2121.dat","wb") as spc_out:
```

```
    while True:
```

```
        what=input("Enter item ")
```

```
        howmany=int(input("Enter quantity "))
```

```
        i=item(what,howmany)
```

```
pickle.dump(i,spc_out,pickle.HIGHEST_PROTOCOL)
del i
ans=input("Add more ? ")
if(ans=='n' or ans=='N'):
    break
```

Enter item covid-19 testing kit

Enter quantity 2500000

Add more ? y

Enter item Corovirus Pneumonia Intensive Care ICU ventilator

Enter quantity 4000

Add more ? n

```
>>> with open("spnotes2121.dat","rb") as spc_in:
```

```
    try:
```

```
        while True:
```

```
            i=pickle.load(spc_in)
```

```
            print(i.nm, " ",i.qy)
```

```
        except EOFError:
```

```
            pass
```

```
covid-19 testing kit 2500000
```

```
Coronavirus Pneumonia Intensive Care ICU ventilator 4000
```

```
>>> with open("spnotes2121.dat","rb") as spc_in:
```

```
    try:
```

```
        while True:
```

```
            i=pickle.load(spc_in)
```

```
            print(i.nm, "\t",i.qy)
```

```
    except EOFError:
```

```
        pass
```

```
covid-19 testing kit 2500000
```

```
Coronavirus Pneumonia Intensive Care ICU ventilator 4000
```

```
>>>
```

Suppose we want to insert a record before any record

```
>>> class item:
```

```
    def __init__(self, name,qty):
```

```
        self.nm=name
```

```
        self.qy=qty
```

```
    def showme(self):
```

```
        print(self.nm, " ",self.qy)
```

```
>>> with open("spnotes2121.dat","rb") as spc_in:
```

```
    try:
```

```
        while True:
```

```
            i=pickle.load(spc_in)
```

```
            print(i.nm, " ",i.qy)
```

```
    except EOFError:
```

```
        pass
```

covid-19 testing kit 2500000

Coronavirus Pneumonia Intensive Care ICU ventilator 4000

We will put one record before the first record (or any record....as we wish)

```
>>> import os
```

```
>>> iii=item("patient GPRS tracker",1000000)
```

```
>>> with open("spnotes21210.dat","wb") as spc_out:
```

```
    with open("spnotes2121.dat","rb") as spc_in:
```

```
        try:
```

```
            while True:
```

```
                ii=pickle.load(spc_in)
```

```
                if("covid-19 testing kit" !=ii.nm):
```

```
                    pickle.dump(ii,spc_out,pickle.HIGHEST_PROTOCOL)
```

For Limited Circulation Only , authored by Somnath PaulChoudhury


```

        elif("covid-19 testing kit" ==ii.nm):

pickle.dump(iii,spc_out,pickle.HIGHEST_PROTOCOL)

        pickle.dump(ii,spc_out,pickle.HIGHEST_PROTOCOL)

    except EOFError:

        pass

>>> spc_in.close()

>>> spc_out.close()

>>> os.remove("spnotes2121.dat")

>>> os.rename("spnotes21210.dat","spnotes2121.dat")

>>> with open("spnotes2121.dat","rb") as spc_in:

    try:

        while True:

            ii=pickle.load(spc_in)

            print(ii.nm," ",ii.qy)

    except EOFError:

        pass

```

patient GPRS tracker 1000000

covid-19 testing kit 2500000

Coronavirus Pneumonia Intensive Care ICU ventilator 4000

For Limited Circulation Only , authored by Somnath PaulChoudhury

The record has been added in the front

PROJECT

SPC autonomous is a car rental company based in West Sikkim. It has recently added to its arsenal 7 world class cars for self driving.

It's open to anyone with an International driving licence and requires appraisal comments from another person of repute for hiring.

As a programmer your task is to develop an automated system using Python.

You should use the following files whose basic structure is given. functionalities you need to decide by yourself.

File name

File 1: Car_mast.dat

Minimum 3 Fields

make

model

rental

add the following records

2019 Telsa Model 3 (here make is Telsa, model is 2019 Model 3, rental give \$ 52 per hour for a minimum of 6 hours)

The remaining records will be as

2020 Volvo XC 60

2019 BMW 5 Series

2019 Cadillac CT6

2020 Lexus LS

2019 Mercedes S - Class

2019 Audi A 8

File 2 rental.dat

tranno (transaction no)

SSNo 8 digit security number (identifying the person)

make (what is rented)

model

hours

billamount

File 3 bank.dat

amount (amount accumulated)

Code development for project idea dated 13/04/2020

```
import pickle

import os

class car:

def __init__(self,make,model,rental):

self.mk=make

self.ml=model

self.rl=rental

def showme(self):

print(self.mk," ",self.ml," ",self.rl)

with open("spcnotes15.dat","wb") as spc_out:

try:

while(True):

mk=input("Input car brand name ")

ml=input("Input car model name ")

rl=int(input("Input rate in USD per hour "))

c=car(mk,ml,rl) # take a note of this step

pickle.dump(c,spc_out,pickle.HIGHEST_PROTOCOL)

ans=input("Wanna Cont? ")

if(ans=='n' or ans=='N'):

break

except EOFError:

pass
```

```
spc_out.close()
```

```
#Now lets read the file
```

```
with open("spcnotes15.dat","rb") as spc_in:
```

```
try:
```

```
while(True):
```

```
cc=pickle.load(spc_in)
```

```
print(cc.mk," ",cc.ml," ",cc.rl)
```

```
except EOFError:
```

```
pass
```

Output is

Input car brand name Telsa

Input car model name 2019 Model 3

Input rate in USD per hour 56

Wanna Cont? y

Input car brand name Volvo

Input car model name 2020 XC 60

Input rate in USD per hour 59

Wanna Cont? y

Input car brand name BMW

Input car model name 2019 BMW 5 Series

Input rate in USD per hour 62

Wanna Cont? y

Input car brand name Cadillac

Input car model name 2019 CT 6

Input rate in USD per hour 67

Wanna Cont? y

Input car brand name Lexus

Input car model name 2020 LS

Input rate in USD per hour 67

Wanna Cont? y

Input car brand name Mercedes

Input car model name 2019 Mercedes S Class

Input rate in USD per hour 75

Wanna Cont? y

Input car brand name Audi

Input car model name 2019 Audi A8

Input rate in USD per hour 100

Wanna Cont? n

Telsa 2019 Model 3 56

Volvo 2020 XC 60 59

BMW 2019 BMW 5 Series 62

Cadillac 2019 CT 6 67

Lexus 2020 LS 67

Mercedes 2019 Mercedes S Class 75

Audi 2019 Audi A8 100

Code development for project idea dated 14/04/2020

Now we can try and develop a menu like this

If we select option 1 we can add more records

1. Add record in Car Master File
2. Display records of Car Master File
3. Update record from Car Master File
4. Rent a Car
5. Return a Car
6. Check Bank Balance
7. Exit

Enter a numerical choice 1..7 1

Input car brand name Ferrari

Input car model name F8 Spider

Input rate in USD per hour 120

Wanna Cont? n

1. Add record in Car Master File
2. Display records of Car Master File
3. Update record from Car Master File
4. Rent a Car
5. Return a Car
6. Check Bank Balance
7. Exit

Enter a numerical choice 1..7 |

If we select option 2 now it shows the record added

```
Enter a numerical choice 1..7 2
Telsa 2019 Model 3 56
Volvo 2020 XC 60 59
BMW 2019 BMW 5 Series 62
Cadillac 2019 CT 6 67
Lexus 2020 LS 67
Mercedes 2019 Mercedes S Class 75
Audi 2019 Audi A8 100
Ferrari F8 Spider 120
1. Add record in Car Master File
2. Display records of Car Master File
3. Update record from Car Master File
4. Rent a Car
5. Return a Car
6. Check Bank Balance
7. Exit
Enter a numerical choice 1..7 |
```

The revised code

```
#Import statements

import pickle

import os

#class definitions

class car:

def __init__(self,make,model,rental):

self.mk=make

self.ml=model

self.rl=rental

def showme(self):

print(self.mk," ",self.ml," ",self.rl)

#main loop

while(True):

print("1. Add record in Car Master File ")

print("2. Display records of Car Master File ")

print("3. Update record from Car Master File ")

print("4. Rent a Car ")

print("5. Return a Car ")

print("6. Check Bank Balance ")

print("7. Exit ")
```

```

# function addinCarMaster()

def addinCarMaster():

with open("spcnotes15.dat","ab") as spc_out:

try:

while(True):

mk=input("Input car brand name ")

ml=input("Input car model name ")

rl=int(input("Input rate in USD per hour "))

c=car(mk,ml,rl) # take a note of this step

pickle.dump(c,spc_out,pickle.HIGHEST_PROTOCOL)

ans=input("Wanna Cont? ")

if(ans=='n' or ans=='N'):

break

except EOFError:

pass

spc_out.close()

# function displayallfromCarMaster()

def displayallfromCarMaster():

with open("spcnotes15.dat","rb") as spc_in:

try:

while(True):

```

```
cc=pickle.load(spc_in)
print(cc.mk," ",cc.ml," ",cc.rl)
except EOFError:
pass

#menu options
ans=int(input("Enter a numerical choice 1..7 "))
if(ans==1):
    addinCarMaster()
elif(ans==2):
    displayallfromCarMaster()
#.....these will be added later
elif(ans==7):
    break
```

Now we can add the remaining functions slowly