

4CS001

## Python Workshop 11: File Handling

### Part 1

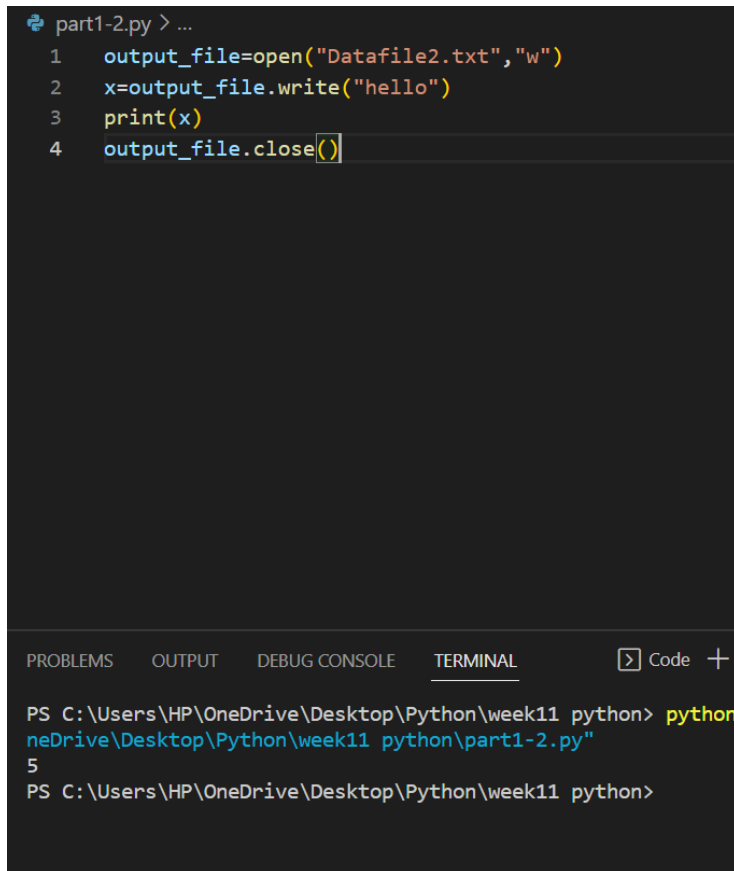
1. Create a program in Python that opens a file named 'datafile.txt' for reading and assigns identifier `input_file` to the file object created.

```
38
39 input_file=open("datafile.txt","r")
40 x=input_file.read()
41 print(x)
42 input_file.close()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
● macbookpro@Avishek-macbook-pro untitle folder % cd "/Users/macbookpro/Desktop/pyt
/usr/bin/python3 "/Users/macbookpro/Desktop/python%
/untitled folder/tot.py"
● macbookpro@Avishek-macbook-pro untitle folder % /usr/bin/python3 "/Users/macbookp
y"
abhishek kc
```

2. Create a program in Python that opens a file named 'datafile2.txt' for writing and assigns identifier `output_file` to the file object created.



The screenshot shows a Python IDE with a dark theme. The top pane displays a script named `part1-2.py` with the following code:

```
1 output_file=open("Datafile2.txt","w")
2 x=output_file.write("hello")
3 print(x)
4 output_file.close()
```

The bottom pane shows the terminal output. The first line is the command prompt `PS C:\Users\HP\OneDrive\Desktop\Python\week11 python>` followed by the command `python neDrive\Desktop\Python\week11 python\part1-2.py`. The second line shows the output `5`. The third line is the command prompt `PS C:\Users\HP\OneDrive\Desktop\Python\week11 python>`.

3. Assume that `input_file` is a file object for a text file open for reading, and `output_file` is a file object for a text file open for writing. Explain the contents of the output after the following code terminates:

```
empty_str = ''
line = input_file.readline()
while line != empty_str:
    output_file.write(line + '\n'
    line = input file.readline()
```

```
t1-3.py > ...
empty_str=""
line=input_file.readline()
while line != empty_str:
    output_file.write(line+"\n")
```

4. Identify the error in the following code:

```
input_file_opened = False
while not input_file_opened:
try:
    file_name = input('Enter file name: ')
    input_file = open(file_name, 'r')
    input_file_opened = True
except: print('Input file not found')
```

=The given code does not close the opened file so here is the code without errors.

```
t1-3.py > ...
input_file_opened=False
while not input_file_opened:
    try:
        file_name=input("Enter file name: ")
        input_file=open(file_name,"r")
        print(input_file.read())
        input_file.close()
        input_file_opened=True
    except:
        print("File not found")
```

## Part 2

1. Write a Python function called `reduce_spaces` that is given a line read from a text file and returns the line with all extra space characters removed:

‘This line has extra space characters’ → ‘This line has extra space characters’

```
1  try:
2      f=open("123.txt","r")
3      x=f.read()
4      y=x.split(" ")
5      print(y)
6      list_1=[]
7      for i in y:
8          list_1.append(i)
9
10     while ("" in list_1:
11         list_1.remove("")
12     print(list_1)
13     stri=""
14     for j in list_1:
15         stri=stri+j+" "
16
17     print(stri)
18 except:
19     print("Invalid name")
```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL Code + -

PS C:\Users\HP\OneDrive\Desktop\Python\week11 python> python -u "c:\OneDrive\Desktop\Python\week11 python\part2-1.py"

['This', 'line', 'has', 'extra', 'space', 'characters\n\n']  
['This', 'line', 'has', 'extra', 'space', 'characters\n\n']  
This line has extra space characters

2. Write a Python function named `extract_temp` that is given a line read from a text file and displays the one number (integer) found in the string:

'The high today will be 75 degrees' → 75.

```
1  try:
2      f=open("123.txt","r")
3      y=f.read()
4      f.close()
5  except:
6      print("Invalid filename")
7
8  def extract_temp(line):
9      x=y.split()
10     for i in x:
11         if (i.isdigit()):
12             print(i)
13         else:
14             pass
15
16  extract_temp(y)
```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\HP\OneDrive\Desktop\Python\week11 python> python -u "C:\Users\HP\OneDrive\Desktop\Python\week11 python\part2-2.py"

75

PS C:\Users\HP\OneDrive\Desktop\Python\week11 python>

3. Write a Python function named `check_quotes` that is given a line read from a text file and returns True if each quote characters in the line has a matching quote (of the same type), otherwise returns False.

'Today's high temperature will be 75 degrees' → False

```
part2-3.py > check_quotes
1  try:
2      f=open("123.txt","r")
3      x=f.read()
4      f.close()
5      print(x)
6  except:
7      print("Invalid file name")
8
9  def check_quotes(line):
10     quote="Today's high temperature will be 75 degrees"
11     x1=quote.split(" ")
12     list_1=[]
13     for i in x1:
14         list_1.append(i)
15     list_2=[]
16     x2=x.split(" ")
17     for j in x2:
18         list_2.append(j)
19     set1=set(list_1)
20     set2=set(list_2)
21     z=set1.issubset(set2)
22     return z
```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL

neDrive\Desktop\Python\week11 python\part2-3.py  
Today's high temperature will be 75 degrees

4. Write a Python function named `count_letters` that is given a line read from a text file and returns a list containing every letter in the line and the number of times that each letter appears (with upper/lower case letters counted together)

'This is a line' → [ ('t', 1), ('h', 1), ('i', 3), ('s', 2), ('a', 1), ('l', 1), ('n', 1), ('e', 1) ]

```
try:
    f=open("123.txt")
    x=f.read()
    f.close()
except:
    print("Invalid Filename")

def count_letters(lines):
    l1=[]
    line=(lines.lower()).split(" ")
    for i in line:
        for k in i:
            l1.append(k)
    l2=[]
    for j in l1:
        y1=str(l1.count(j))
        y2=str((j,y1)).strip(" ")    ##
        y3=j.replace(j,y2) #or y3=j.replace(j, "("+"\""+j+"\""+", "+y1+")")
        l2.append(y3)
    l3=[]
    l3=list(dict.fromkeys(l2))
    l4=str(l3).replace("'", "")
    print(l4)
count_letters(x)
```

```
PS C:\Users\HP\OneDrive\Desktop\Python\week11 python> python -u "c:\Users\HP\OneDrive\Desktop\Python\week11 python\part2-4.py"
[('t', '1'), ('h', '1'), ('i', '3'), ('s', '2'), ('a', '1'), ('l', '1'), ('n', '1'), ('e', '1'), ('\\n', '2')]
PS C:\Users\HP\OneDrive\Desktop\Python\week11 python>
```

5. Write a Python function named `interleave_chars` that is given two lines read from a text, and returns a single string containing the characters of each string interleaved: 'Hello', 'Goodbye' → 'HGeololdobye'

```
part2-5.py 7 ...
1  try:
2      f=open("123.txt")
3      a=f.readline()
4      b=f.readline()
5      f.close()
6  except:
7      print("Invalid filename")
8
9  def interleave_chars(line1,line2):
10     a1=[]
11     b1=[]
12     for i in line1.strip("\n"):
13         a1.append(i)
14     a1.append("")
15     a1.append("")
16     for j in line2.strip("\n"):
17         b1.append(j)
18     for k in range(0,len(b1)):
19         z=a1[k]+b1[k]
20         print(z,end="")
21     interleave_chars(a,b)
```

PROBLEMS 3 OUTPUT TERMINAL ... Code + - [ ] [X] ...

```
PS C:\Users\HP\OneDrive\Desktop\Python\week11 python> python
\Users\HP\OneDrive\Desktop\Python\week11 python\part2-5.py"
'H'e'llo
PS C:\Users\HP\OneDrive\Desktop\Python\week11 python>
```



6. Give a for loop that counts all the characters in a string assigned to variable `line`, except blanks and the newline character.

```
1  try:
2      f=open("123.txt")
3      x=[]
4      for i in f:
5          y1=i.strip("\n")
6          for j in y1:
7              x.append(j)
8      while " " in x:
9          x.remove(" ")
10     f.close()
11     l2=[]
12     for i in x:
13         count_1=(x.count(i))
14         l2.append((i,count_1))
15     l3=list(dict.fromkeys(l2))
16     print(l3)
17 except:
18     print("invalid filename")
```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\HP\OneDrive\Desktop\Python\week11 python> python -u "c:\Users\HP\OneDrive\Desktop\Python\week11\part2-6.py"

[("'", 6), ('A', 1), ('k', 1), ('r', 1), ('i', 2), ('t', 1), ('D', 1), ('e', 1), ('v', 1)]

PS C:\Users\HP\OneDrive\Desktop\Python\week11 python>

7. For variable `month` which contains the full name of any given month, give an expression to display just the first three letters of the month.

```
part2-7.py > ...
1  try:
2      f=open("123.txt")
3      a=f.read()
4      f.close()
5      print(a[0:3])
6  except:
7      print("invalid filename")
```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\HP\OneDrive\Desktop\Python\week11 pyt
hon\part2-7.py"
feb
PS C:\Users\HP\OneDrive\Desktop\Python\week11 pyt
```

8. Give an expression that displays `True` if the letter 'r' appears in a given `month` name stored in variable `month`, otherwise displays `False`.

```
part2-8.py > ...
1 month=input("Enter a month: ")
2 if ("r" in month):
3     print("True")
4 else:
5     print("False")
```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\HP\OneDrive\Desktop\Python\week11
non\part2-8.py"
Enter a month: March
True
PS C:\Users\HP\OneDrive\Desktop\Python\week11
```

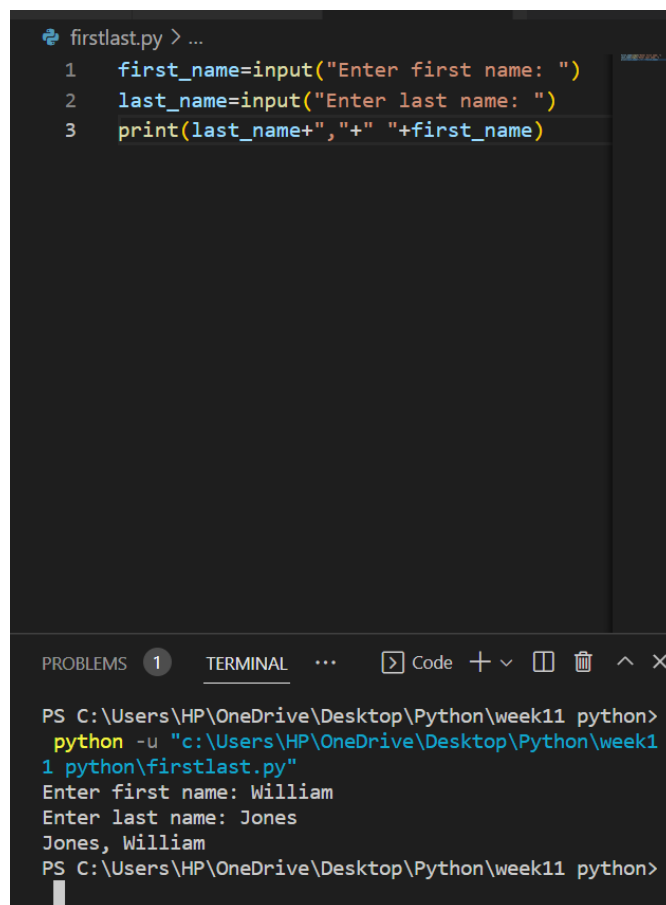
9. Give an expression for determining how many times the letter 'r' appears in a given month name stored in variable `month`.

```
1 month=input("Enter a month: ")
2 x=month.count("r")
3 print(x)
```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\HP\OneDrive\Desktop\Python\week 2\part2-9.py  
Enter a month: February  
2  
PS C:\Users\HP\OneDrive\Desktop\Python\week 2\part2-9.py

10. For a person's first name stored in variable `first_name`, and last name stored in variable `last_name`, give an expression that displays the person's name formatted exactly as follows: Jones, William.



```
firstlast.py > ...
1 first_name=input("Enter first name: ")
2 last_name=input("Enter last name: ")
3 print(last_name+", "+" "+first_name)
```

PROBLEMS 1 TERMINAL ... Code + - [] [X] ^ X

```
PS C:\Users\HP\OneDrive\Desktop\Python\week11 python>
python -u "c:\Users\HP\OneDrive\Desktop\Python\week1
1 python\firstlast.py"
Enter first name: William
Enter last name: Jones
Jones, William
PS C:\Users\HP\OneDrive\Desktop\Python\week11 python>
```

11. Give an instruction that determines if a given social security number represented as a string and stored in variable `ss_num`, contains any non- digits.

```
numdigit.py 7 ...
1  ss_num=input("Enter social security number: ")
2  new=filter(str.isdigit,ss_num)
3  new2="".join(new)
4  if (ss_num==new2):
5      print("No non-digits")
6  else:
7      print("YES! Non-digits are present")
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL Code + -

```
PS C:\Users\HP\OneDrive\Desktop\Python\week11 python> python -u "c:\Users\HP\OneDrive\Desktop\Python\week11 python\numdigit.py"
Enter social security number: 0,1,2,3,4,5
YES! Non-digits are present
PS C:\Users\HP\OneDrive\Desktop\Python\week11 python> 
```

12. Give an instruction that determines the index of the '@' character in an email address stored in variable `email_addr`.

```
38
39 email_addr=input("enter email_address :")
40 if ("@" in email_addr):
41     index_position=email_addr.index("@")
42     print(index_position)
43 else:
44     print("not peresent")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Python + - [ ] [X] [Y]

- macbookpro@Avishek-macbook-pro untitle folder % /usr/bin/python3 "/Users/macbookpro/Desktop/python/untitled folder/tot.py"  
y"  
abhishek kc
- macbookpro@Avishek-macbook-pro untitle folder % cd "/Users/macbookpro/Desktop/python/untitled folder"  
/usr/bin/python3 "/Users/macbookpro/Desktop/python/untitled folder/tot.py"
- macbookpro@Avishek-macbook-pro untitle folder % /usr/bin/python3 "/Users/macbookpro/Desktop/python/untitled folder/tot.py"  
enter email\_address :kcabhishek@gmail.com  
10

13. For a variable named `date` containing a date in the form 12/14/2012, give an expression that replaces all slashes characters with dashes.

```
date.py > ...
1  date=input("Enter date: ")
2  x=date.replace("/", "-")
3  print(x)
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL Code + -

```
PS C:\Users\HP\OneDrive\Desktop\Python\week11 python> python -u "c:
sers\HP\OneDrive\Desktop\Python\week11 python\date.py"
Enter date: 02/02/2023
02-02-2023
PS C:\Users\HP\OneDrive\Desktop\Python\week11 python> |
```



14. For a variable named `err_mesg` that contains error messages in the form `**error message **`, give an expression that produces a string containing the error message without the leading and trailing asterisks and blank characters.

```
err_mesg=input("Enter error message: ")
x=err_mesg.strip("**")
y=x.strip(" ")
print(y)
```

## Part 3

1. Write a program that opens and reads a text file and displays how many lines of text are in the file.

```
part3.py
1  try:
2      f=open("123.txt")
3      #x=f.read()
4      count_lines=0
5      l1=[]
6      for i in f:
7          l1.append(i)
8
9      print(len(l1))
10
11 except:
12     print("Invalid filename")
```

PROBLEMS 1 TERMINAL ... Code + - [] [X] ^

```
PS C:\Users\HP\OneDrive\Desktop\Python\week11 python> python -u "c:\Users\HP\OneDrive\Desktop\Python\week11 python\part3.py"
6
PS C:\Users\HP\OneDrive\Desktop\Python\week11 python>
```

2. Write a program that reads a text file named `original_text`, and writes every other line, starting with the first line, to a new file named `new_text`.

```
try:
    f=open("original_text.txt")
    f1=open("new_text.txt","w")
    for i in f:
        f1.write(i)
    f.close()
    f1.close()
except:
    print("Invalid filename")
```

3. Write a program that reads a text file named `original_text`, and counts how many time the letter 'e' occurs (the most frequently occurring letter in English), and displays how many occurrences there are.

```

1  try:
2      f=open("original_text.txt")
3      l=[]
4      for i in f:
5          l.append(i)
6      l1=str(l).strip("\n")
7      count=0
8      for j in l1:
9          for k in j:
10             if (k=="e"):
11                 count=count+1
12      f.close()
13      print("The number of times e is present in the file is: "+str(count))
14  except:
15      print("Invalid filename")

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\HP\OneDrive\Desktop\Python\week11 python> python -u "c:\Users\HP\OneDrive\Desktop\Python\week11 python\part3-3.py"

The number of times e is present in the file is: 2

PS C:\Users\HP\OneDrive\Desktop\Python\week11 python>

4. Write a program that reads a text file containing numerical expressions on each line and print them out along with the results. For example, for the numerical expression  $4 + 2$  in your file, your program should output:  $4 + 2 = 6$ .

```
part3-4.py > ...
1  try:
2      f=open("original_text.txt")
3      l=[]
4      for i in f:
5          for j in i:
6              l.append(j)
7      summ=int(l[0])+int(l[2])
8      print(summ)
9
10 except:
11     print("Invalid filename")
```

PROBLEMS 1 OUTPUT TERMINAL ... Code + - [ ] [ ] ^ x

```
PS C:\Users\HP\OneDrive\Desktop\Python\week11 python> pyt
hon -u "c:\Users\HP\OneDrive\Desktop\Python\week11 python
\part3-4.py"
6
PS C:\Users\HP\OneDrive\Desktop\Python\week11 python>
```

## Part 4 (Optional)

Write a Python program that encrypts and decrypts text files using a substitution cipher. Your program should ask the user for the name of a text file and whether they would like to encrypt or decrypt. Once the process is complete, you should write the output to a new text file with a modified name:

```
This program will encrypt and decrypt text files
```

```
Enter (e) to encrypt a password, and (d) to decrypt: e
```

```
Enter the name of a text file to encrypt: hello.txt
```

```
Output written to: encrypted_hello.txt
```

Your program should catch exceptions and print helpful error messages. You should use your solution to Coding Challenge 03 to help you.