



INDIAN SCHOOL AL WADI AL KABIR

Class: XII Comp. Sci.	Department: Computer Science	Date of submission: 16/05/2021
Worksheet No: 3	Topic: File Handling	Note: Write in your class note book. Execute programs

Answer the following

Section A

1. _____ in Python are interpreted as a sequence or stream or stream of bytes stored on some storage media. **Ans: Files**
2. _____ method writes a list of strings to a file. **Ans: writelines()**
3. The _____ method of a file object flushes any unwritten information and closes the file object. **Ans: close()**
4. The _____ method is used to change the file name or folder name and _____ method is used to remove a file. **Ans: os.rename() os.remove()**
5. The read() function reads data from the _____ of a file. **Ans: Beginning**
6. The pickle module produces two main methods _____ and _____ for writing and reading operations. **Ans: dump() and load()**
7. The readlines() return a list of lines from the file till _____. **Ans: EOF (End of File)**
8. The _____ method reads 'n' characters from the file. **Ans: read(n)**
9. _____ function is used to force transfer of data from buffer to file. **Ans: flush()**
10. The default file open mode is _____. **Ans: read**
11. Opening a file in append mode will place the file pointer at _____ position. **Ans: EOF**
12. A text file stores in _____ or _____ characters. **Ans: ASCII, Unicode**
13. A _____ defines the type of operations that is to be performed on the file. **Ans: File Mode**
14. _____ function returns a list of strings, each separated by "\n". **Ans: readlines()**
15. _____ statement is used to open a file C:\test.txt for reading. **Ans: f1=open("C:\test.txt", "r") (Note: \t indicates tab space. Should use double \\)**
16. _____ statement is used to read two characters from a file object fobj. **Ans: fobj.read(2)**
17. _____ statement is used to read the next line of the file from a file object fobj. **Ans: fobj.readline()**
18. _____ statement is used to read the remaining lines of the file from file object fobj. **Ans: fobj.readlines()**
19. The readlines() method returns _____. **Ans: A list of lines**
20. _____ module is required to use the built-in-function dump(). **Ans: pickle**

Section B

1. What is the difference between readline() and readlines() function?

The readline() function reads from a file in read mode and returns the next line in the file or a blank string if there are no more lines. (Returned data is string type)

The readlines() function also reads from a file in read mode and returns a list of all lines in the file. (Returned data is of list type).

2. Write a single loop to display all the contents of a text file “sample.txt” after removing leading and trailing whitespaces.

```
for line in open(“sample.txt”):  
    print(line.strip())
```

3. Differentiate between file modes r+ and w+ with respect to Python.

r+ opens a file for both reading and writing. The file pointer placed at the beginning of the file.

w+ opens a file for both reading and writing. Overwrites the existing file if the file exists. If the file does not exist, creates a new file for reading and writing.

4. Differentiate between file modes r+ and rb+ with respect to Python.

r+ opens a file for both reading and writing. The file pointer placed at the beginning of the file.

rb+ opens a file for both reading and writing in binary format. The file pointer is placed at the beginning of the file.

5. What is pickling and unpickling?

Pickle module uses dump() method to store the data object (sequence / numeric type) into a specific file. It is known as pickling. It uses load() method to retrieve the data object from the file. It is known as unpickling.

Section C

1. What is the output of following code:

```
F1 = open(“demo.txt”, “r”)  
Size = len(F1.read())  
print(F1.read(10))
```

Output: No Output.

2. Write a statement in Python to open a text file “MyBook.txt” in read mode.

```
File1=open(“MyBook.txt”,”r”)
```

3. Write a statement in Python to open a text file "MyLibrary.txt" in read and write mode.

```
File1=open("MyLibrary.txt","r+")
```

4. Write a statement in Python to open a text file "MyNote.txt" in write mode.

```
File1=open("MyNote.txt","w")
```

5. Write a statement in Python to open a text file "MyBook.txt" in append and read mode.

```
File1=open("MyBook.txt","a+")
```

6. Write a statement in Python to open a binary file "Textiles.dat" in read mode.

```
File1=open("Textiles.dat","rb")
```

7. Write a statement in Python to open a binary file "College.dat" in read and write mode

```
File1=open("College.dat","rb+")
```

8. Write a statement in Python to open a binary file "Address.dat" in append mode.

```
File1=open("Address.dat","ab")
```

Section D

1. Write a function disp_Long() that accepts a file name and display the longest line of the file.

```
def longline(filename):  
    long = ""  
    for line in open(filename):  
        if len(line) > len(long):  
            long = line  
    print("Longest Line : ",long)  
    print("Length of the Longest line = ", len(long))  
longline("demo.txt")
```

2. Write a method in Python to write multiple lines of text contents into a text file Diary.txt

```
def WriteMulLines():  
    f=open("Diary.txt",'w')  
    while True:  
        line=input("Enter Line : ")  
        f.writelines(line)  
        f.writelines("\n")  
        ch=input("Want to write More lines (Y/N)? ")  
        if ch.upper()=='N':  
            break  
    f.close()  
WriteMulLines()
```

3. Write a user defined function in Python that displays the number of lines starting with 'S' in the file "Student.txt"

```
def countLineStartS():  
    f=open("Student.txt","r")  
    c = 0  
    lines = f.readlines()
```

```

for k in lines:
    if k[0] == 'S':
        print("Line : ",k)
        c = c + 1
print("No. of Lines start with S = ", c)
countLineStartS()

```

4. Consider a binary file Product.dat containing details such as Prodno:Prodname:Price (separator (:). Write a Python function to display details of those products with priced more than 2500.

```

def CheckPrice():
    f = open("Product.dat", "r")
    line = f.readline()
    while(line):
        p = line.split(':')
        if float(p[2]) > 2500:
            print(line)
        line = f.readline()
    f.close()
CheckPrice()

```

5. Write a function countWord() in Python to read the text file "Demo.txt" and count the number of times 'We' occurs in the file.

```

def countWe():
    f=open("demo.txt", "r")
    c = 0
    x = f.read()
    word = x.split()
    for k in word:
        if k == "We":
            c = c + 1
    print("No of word We = ",c)
    f.close()
countWe()

```

6. Write a method in Python to read lines from a text file "Project.txt" and displays those lines which start with the alphabet 'I'.

```

def countLineStartI():
    f=open("Project.txt", "r")
    lines = f.readlines()
    for k in lines:
        if k[0] == 'I':
            print("Line : ",k)
countLineStartI()

```

7. Write a program to display all the records in a file along with line/record number.

```
f = open("Seminar.txt","r")
count = 0
x = ""
while True:
    x = f.readline()
    if x == "":
        break
    count = count + 1
    print(count, x)
f.close()
```

8. Write code to print just the second last line of a text file 'Test.txt'.

```
fin = open("Test.txt","r")
lines = fin.readlines()
fin.close()
print("Second Last Line = ",lines[-2])
```

9. Write a function in Python to count the number of lines in a text file 'Garden.txt' which are starting with the alphabet 'S'.

```
def countlinesS():
    f1 = open("Garden.txt","r")
    lines = f1.readlines()
    c = 0
    for line in lines:
        if line[0].upper() == 'S':
            c = c + 1
    print("No. lines start with S = ", c)
    f1.close()
countlinesS()
```

10. Write a method DisplayWords() in Python to read lines from a text file BOOK.txt and display those words which are less than 5 characters.

```
def dispWords():
    f1 = open("Book.txt","r")
    lines = f1.readlines()
    print(lines)
    word = str(lines).split()
    print(word)
    for w in word:
        if len(w) < 5:
            print(w)
    f1.close()
dispWords()
```

11. Write a program to display the size of file after removing EOL characters, leading and trailing white spaces and blank lines.

```
myfile=open("demo.txt","r")
s1=" "
size=0
tsize=0
while s1:
    s1=myfile.readline()
    tsize = tsize + len(s1)
    size = size + len(s1.strip())
print("Size of file after removing all EOL characters & Blank links: ",size)
print("Total size of the file = ",tsize)
myfile.close()
```

12. Write a program that copies a text file “Source.txt” onto “Target.txt” barring the lines starts with “@” sign.

```
def filter(oldfile, newfile):
    fin = open(oldfile,"r")
    fout = open(newfile, "w")
    while True:
        text=fin.readline()
        if len(text) == 0:
            break
        if text[0] == '@':
            continue
        fout.write(text)
    fin.close()
    fout.close()
filter("Source.txt","Target.txt")
```