File Handling

The key function for working with files in Python is the open(arg1,agr2) function.

The open() function takes two parameters; *filename,* and *mode.*

There are four different methods (modes) for opening a file:

"r" - Read - Default value. Opens a file for reading, error if the file does not exist

"a" - Append - Opens a file for appending, creates the file if it does not exist

"w" - Write - Opens a file for writing, creates the file if it does not exist

"x" - Create - Creates the specified file, returns an error if the file exists

In addition you can specify if the file should be handled as binary or text mode

"t" - Text - Default value. Text mode

"b" - Binary - Binary mode (e.g. images)

Syntax

To open a file for reading it is enough to specify the name of the file:

f = open("demofile.txt",”x”)

The code above is the same as:

f = open("demofile.txt", "rt")

Because "r" for read, and "t" for text are the default values, you do not need to specify them.

Create a New File

To create a new file in Python, use the open() method, with one of the following parameters:

"x" - Create - will create a MLSU, returns an error if the MLSU exist

"a" - Append - will create a MLSU if the specified MLSU does not exist

"w" - Write - will create a MLSU if the specified file does not exist

Delete a File

To delete a file, you must import the OS module, and run its os.remove(filename) function:

Example

Remove the file "demofile.txt":

```
import os
os.remove("demofile.txt")
```

Check if File exist:

To avoid getting an error, you might want to check if the file exists before you try to delete it:

Example

Check if file exists, *then* delete it:

```
import os
if os.path.exists("demofile.txt"):
  os.remove("demofile.txt")
else:
  print("The file does not exist")
```

Delete Folder

To delete an entire folder, use the os.rmdir() method:

Example

Remove the folder "myfolder":

```
import os
os.rmdir("myfolder")
```

example 1: To create file with user input

```
text_file = open('/desktop/file.txt','w')

word_list= []

for i in range (1, 5):
    print("Please enter data: ")
    line = input()
    word_list.append(line)


text_file.writelines(word_list)
text_file.close( )
```

Replacing a word in file

```
with open('file.txt', 'r+') as f:

    #read file
    file_source = f.read()

    #replace 'PHP' with 'PYTHON' in the file
    replace_string = file_source.replace('PHP', 'PYTHON')

    #save output
    f.write(replace_string)
```

Another way:

```
input_data = open('input.txt','r')
output_data = open('output.txt','w')
x = input('Enter the word to be replaced:')
y = input('What should I replace all occurrences of ', x, 'with?')

for line in input_data.split():
    for word in line:
        if word in line:
            if word==x:
                  output_data.replace(x,y)
    output_data.write(word)
input_data.close()
output_data.close()
```

More programs on files

Q. 1 Write a function in python to read the content from a text file "poem.txt" line by line and display the same on screen.

Source Code

```
def read_file():
    file = open("poem.txt","r")
    for line in file:
        print(line, end="")
    file.close()

read_file()
```

Q.2 Write a function in Python to count and display the words words in a text file.

Source Code

```
def count_words():
    file = open("notes.txt","r")
    count = 0
    for line in file:
        words = line.split()
        for word in words:
            count += 1
    print("Total words are",count)

count_words()
```

Q.3 A text file named "matter.txt" contains some text, which needs to be displayed such that every next character is separated by a symbol "#".

Write a function definition for hash_display() in Python that would display the entire content of the file matter.txt in the desired format.
Example :
If the file matter.txt has the following content stored in it :
input THE WORLD IS ROUND

The function hash_display() should display the following content :
output T#H#E# #W#O#R#L#D# #I#S# #R#O#U#N#D#

Source Code

```
def count_hash():
    file = open("matter.txt","r")
    data = file.read()
    for letter in data:
        print(letter, end="#")

    file.close()

count_hash()
```

Q.4 Aditi has used a text editing software to type some text. After saving the article as WORDS.TXT, she realised that she has wrongly typed alphabet J in place of alphabet I everywhere in the article.

Write a function definition for JTOI() in Python that would display the corrected version of entire content of the file WORDS.TXT with all the alphabets "J" to be displayed as an alphabet "I" on screen.
Note: Assuming that WORDS.TXT does not contain any J alphabet otherwise.

Example:
If Aditi has stored the following content in the file WORDS.TXT:

WELL, THJS JS A WORD BY JTSELF. YOU COULD STRETCH THJS TO BE A SENTENCE

The function JTOI() should display the following content:
WELL, THIS IS A WORD BY ITSELF. YOU COULD STRETCH THIS TO BE A SENTENCE

Source Code

```
def JTOI():
    file = open("words.txt","r")
    data = file.read()
    for letter in data:
        if letter == 'J':
            print("I",end="")
        else:
            print(letter,end="")

    file.close()

JTOI()
```

Important Q.5 A binary file "Book.dat" has structure [BookNo, Book_Name, Author, Price].
i. Write a user defined function createFile() to input data for a record and add to Book.dat.
ii. Write a function countRec(Author) in Python which accepts the Author name as parameter and count and return number of books by the given Author are stored in the binary file "Book.dat"

Source Code

```
import pickle

def createFile():
    file = open("book.dat","ab")
    BookNo = int(input("Enter book number: "))
    Book_Name = input("Enter book Name: ")
    Author =input("Enter author: ")
    Price = int(input("Enter price: "))
    record = [BookNo, Book_Name, Author, Price]
    pickle.dump(record, file)
    file.close()

def countRec(Author):
    file = open("book.dat","rb")
    count = 0
    try:
        while True:
            record = pickle.load(file)
            if record[2]==Author:
                count+=1
    except EOFError:
        pass
    return count
    file.close()

#To test working of functions
def testProgram():
    while True:
        createFile()
        choice = input("Add more record (y/n)? ")
        if choice in 'Nn':
            break
    Author = input('Enter author name to search: ')
    n = countRec(Author)
    print("No of books are",n)

testProgram()
```

Q.6 A binary file "STUDENT.DAT" has structure (admission_number, Name, Percentage). Write a function count_rec() in Python that would read contents of the file "STUDENT.DAT" and display the details of those students whose percentage is above 75. Also display number of students scoring above 75%

Source Code

```
def count_rec():
    file = open("STUDENT.DAT","rb")
    count = 0
    try:
        while True:
            record = pickle.load(file)
            if record[2] > 75:
                print(record)
                count+=1
    except EOFError:
        pass
    print('No of students having more than 75% are',
count)
    file.close()
```

Q.7 A binary file players.dat, containing records of following list format: [code, name, country and total runs]

(i)Write a python function that display all records where player name starts from 'A' (ii)Write a python function that accept country as an argument and count and display the number of players of that country.

(iii)Write a python function that add one record at the end of file.

Source Code

```
import pickle

def createFile():
    file = open("players.dat","ab")
    Code = int(input("Enter player code: "))
    Name = input("Enter player Name: ")
    Country =input("Enter player country: ")
    Total_Runs = int(input("Enter total runs of player: "))
    record = [Code, Name, Country, Total_Runs]
    pickle.dump(record, file)
    file.close()

def search():
    file = open("players.dat","rb")
    try:
        while True:
            record = pickle.load(file)
            if record[1][0] == 'A':
                print(record)
    except EOFError:
        pass
    file.close()
```

```
def countRec(Country):
    file = open("players.dat","rb")
    count = 0
    try:
        while True:
            record = pickle.load(file)
            if record[2]==Country:
                count+=1
    except EOFError:
        pass
    return count
    file.close()

def testProgram():
    while True:
        createFile()
        choice = input("Add more record (y/n)? ")
        if choice in 'Nn':
            break
    print("the number of players whose name starts with A:",search())
    Country = input('Enter country name to search: ')
    n = countRec(Country)
    print("No of players are",n)

testProgram()
```

Q.8 A binary file school.dat has structure(rollno, name, class, fees)

Write a definition for function total_fees( ) that reads each object of file and calculate the total fees of students and display the same.

Source Code

```
def total_fees():
    file = open("school.dat","rb")
    try:
        total = 0
        while True:
            record = pickle.load(file)
            total += record[3]:
    except EOFError:
        pass
    print('Total Fees: ',total)
    file.close()
```

Q.9 Reverse words in a file

```
# Open file in read mode
f = open('GFG.txt', 'r')

lines = f.readlines()

# Close file
f.close()

# User's choice
choice = 1
```

```
line = lines[choice].split()

# line is reversed
Reversed = " ".join(line[::-1])

lines.pop(choice)
lines.insert(choice, Reversed)

# Open file in write mode
u = open('GFG.txt', 'w')

# Write the new content in file
# and note, it is overwritten
u.writelines(lines)
u.close()
```

Q.10 Program using pickle module to be used in files

# Pickle Module translates an in-memory Python object into a serialized byte stream—a string of bytes that can be written to any file-like object.

```
import pickle

def storeData():
    # initializing data to be stored in db
    Omkar = {'key' : 'Omkar', 'name' : 'Omkar Pathak',
    'age' : 21, 'pay' : 40000}
    Jagdish = {'key' : 'Jagdish', 'name' : 'Jagdish Pathak',
    'age' : 50, 'pay' : 50000}

    # database
    db = {}
    db['Omkar'] = Omkar
    db['Jagdish'] = Jagdish

    # Its important to use binary mode
    dbfile = open('examplePickle', 'ab')

    # source, destination
    pickle.dump(db, dbfile)
    dbfile.close()

def loadData():
    # for reading also binary mode is important
    dbfile = open('examplePickle', 'rb')
    db = pickle.load(dbfile)
    for keys in db:
        print(keys, '=>', db[keys])
    dbfile.close()

if __name__ == '__main__':
    storeData()
    loadData()
```