**DEPARTMENT OF INFORMATION TECHNOLOGY**

**COURSE CODE:** DJ19ITL406

**COURSE NAME:** Programing Laboratory 2 (Python)  **CLASS:** SYBTECH

**EXPERIMENT NO. 10**

**CO/LO: CO1, CO2.**

**AIM / OBJECTIVE:**

Write python programs to understand different File handling operations with exception handling.

**DESCRIPTION OF EXPERIMENT:**

1. File handling modes
2. Three types of exception

**QUESTIONS:**

1. WAP to accept name and roll number of students and store it in file. Read and display the stored data. Accept/ display should be done using menu. Previous data should be retained and new data shouldbeappendedattheEOF.

**SOURCE CODE:**

exit = False

while exit == False:

n = input("1)append data\n2)Show data\n3)exit\n")

if n=='1':

f = open('Myfile.txt','a')

name = input('Enter your name : ')

rollNo = input('Enter you roll no : ')

f.write(name)

f.write(rollNo)

f.close()

print("\n")

elif n=='2':

f = open('Myfile.txt','r')

print("\n")

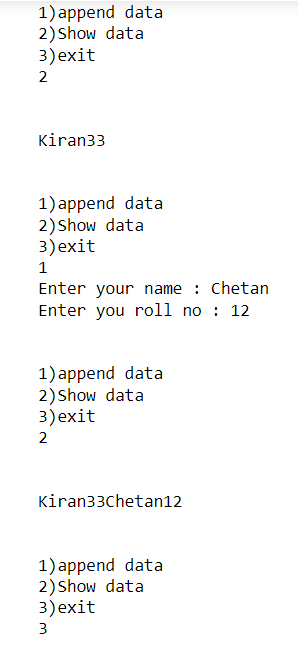
print(f.read())

print("\n")

else:

exit = True

**OUTPUT:**

****

1. Alsocheckiffileexistsornot(pickle)

**SOURCE CODE:**

#same code with pickle:

import pickle

import os

exit = False

while exit == False:

n = input("1)append data\n2)Show data\n3)exit\n")

if n=='1':

if os.path.exists('Myfile.pickle')==True:

f = open('Myfile.pickle','wb')

name = input('Enter your name : ')

rollNo = input('Enter you roll no : ')

pickle.dump(name,f)

pickle.dump(rollNo,f)

f.close()

else:

print("Creating a new file!")

f = open('Myfile.pickle','wb')

name = input('Enter your name : ')

pickle.dump(name,f)

f.close()

print("\n")

elif n=='2':

f = open('Myfile.pickle','rb')

data = pickle.load(f)

print("\n")

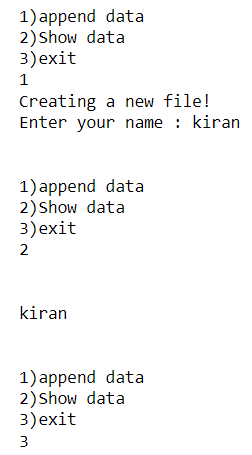
print(data)

print("\n")

else:

exit = True

**OUTPUT:**

****

1. WAP to copy contents of 1 file to another after converting text to same case (upper/ lower). Let userspecifynameof source and destinationfiles.

**SOURCE CODE:**

data = input("Enter data")

SfileName = input("Enter source file name")

DfileName = input("Enter destination file name")

f1 = open(SfileName,'a')

f2 = open(DfileName,'a')

f1.write(data)

f1.close()

f1 = open(SfileName,'r')

for line in f1:

line = line.lower()

f2.write(line)

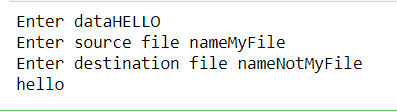
f2.close()

f2 = open(DfileName,'r')

print("The data stored in the secod file is : ")

print(f2.read())

**OUTPUT:**

****

1. Make your exception class“InvalidMarks” which is thrown when marks obtained by student exceeds100

**SOURCE CODE:**

class invalidMarks(Exception):

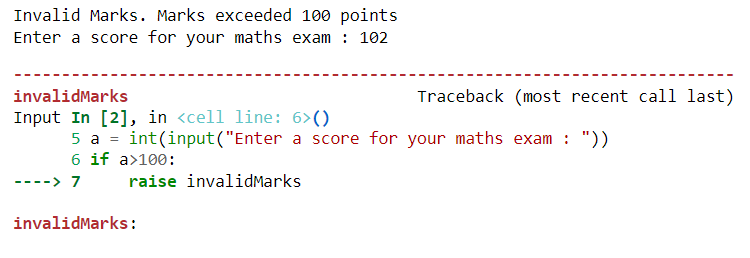
print("Invalid Marks. Marks exceeded 100 points")

a = int(input("Enter a score for your maths exam : "))

if a>100:

raise invalidMarks

**OUTPUT:**



1. WAP that accepts the values of a, b, c and d. Calculate and display ((a+d) + (b\*c))/ (b\*d).

**SOURCE CODE:**

a = int(input("Enter value for a :"))

b = int(input("Enter value for b :"))

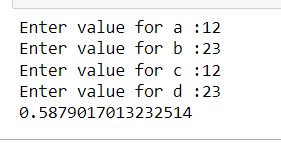
c = int(input("Enter value for c :"))

d = int(input("Enter value for d :"))

num = ((a+d) + (b\*c))/(b\*d)

print(num)

**OUTPUT:**

****

1. Createuserdefinedexceptiontodisplaypropermessagewhenvalueof(b\*d)iszero 3. MakeuseofassertstatementtocatchAssertion Error

**SOURCE CODE:**

class invalidNumbers(Exception):

print("b or d cannot be zero")

a = int(input("Enter value for a :"))

b = int(input("Enter value for b :"))

c = int(input("Enter value for c :"))

d = int(input("Enter value for d :"))

if(b==0 or d==0):

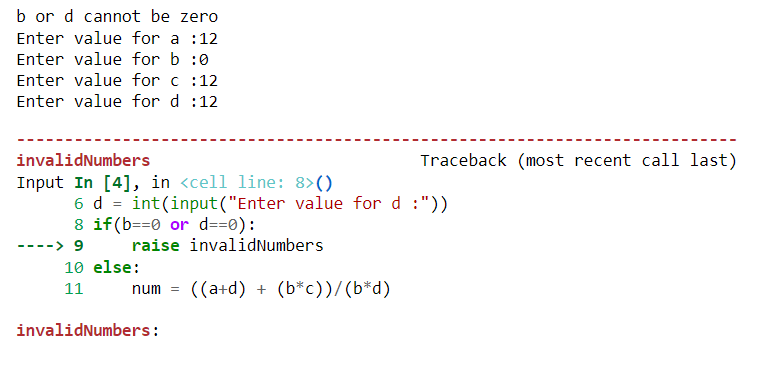
raise invalidNumbers

else:

num = ((a+d) + (b\*c))/(b\*d)

print(num)

**OUTPUT:**

****

**OBSERVATIONS / DISCUSSION OF RESULT:**

After performing the experiment, we observed that

1)Python can be used to create, read append and write into a file.

2)The key difference between append and write methods in python is that append attaches the given data into file without changing the data in the files while write overwrites the data present in the file with data specified by the user.

3)User can raise a custom exception with the keyword raise

4)In order to make user defined exception class, the class must extend the Exception class

**CONCLUSION:**

Hence, we successfully implemented

1)Different File handling modes

2)Different ways to handle exceptions

**REFERENCES:**

**Website References:​**

[1] https://www.w3schools.com/python