**Question: Write a program to read lines of input from file, process the data, and out the results to a file.**

1. Plan.txt

Program will have:

Main Method:

Global variables:

. employeeDetail dictionary

. customerDetail dictionary

. tranactionDetail dictionary

. transactionLog dictionary

Functions:

1. ReadDataOfFile function

- Read whole content of the data.txt

2. CreateEmployeeDictionary function

- Append data of employee read from the file data.txt to employeeDetail dictionary.

3. CreateCustomerDictionary function

- Append data of customer read from the file data.txt to customerDetail dictionary.

4. CreateTransactionDictionary function

- Append transaction data read from the file data.txt to transactionDetail dictionary.

5. Transaction function

- Process deposit and withdraw

6. Deposit function

- Read customer details and employee details.

- Calculate new account balance after deposit.

- Append deposited transaction details to the transactionLog dictionary

7. Withdraw function

- Read customer details and employee details.

- Calculate new account balance after withdraw.

- Append withdrawn transaction details to the transactionLog dictionary

8. WriteTransactionLogToFile Function

-Write 123456789012345678901234567890123456789012345678901234567890 in the first line of file

- Write the data of transactionLog dictionary in format:

customerName employeeName deposited(+)/withdrawn(-)$Amount $ customerNewAccountBalance

Task, Time, LinesOfCode

Main Method, 15, 15

ReadDataOfFile function, 5, 3

CreateEmployeeDictionary function, 10, 15

CreateCustomerDictionary function, 10, 15

CreateTransactionDictionary function, 15, 15

Transaction function, 5, 8

Deposit function, 20, 20

Withdraw function, 20, 20

WriteTransactionLogToFile function, 10, 5

Error Fixing, 20, 0

1. Task.log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Timestamp | Activity | Status | Duration | Lines of Code |
| 06-10-2020-11:49 | Coding | Writing Main Method | 6 | 11 |
| 06-10-2020-11:57 | Coding | Writing ReadDataOfFile function | 3 | 1 |
| 06-10-2020-12:01 | Coding | Writing CreateEmployeeDictionary function | 7 | 5 |
| 06-10-2020-12:08 | Coding | Writing CreateCustomerDictionary function | 3 | 7 |
| 06-10-2020-12:11 | Coding | Writing CreateTransactionDictionary function | 3 | 9 |
| 06-10-2020-12:14 | Coding | Writing Transaction function | 2 | 6 |
| 06-10-2020-12:16 | Coding | Writing Deposit function | 11 | 4 |
| 06-10-2020-12:28 | Coding | Writing Withdraw function | 8 | 4 |
| 06-10-2020-13:33 | Coding | Writing WriteTransactionLogToFile function | 5 | 3 |
| 06-10-2020-13:40 | Coding | Error fixing | 8 | - |
| 06-10-2020-13:36 | Coding | Testing | 17 | - |

1. Defect.log

|  |  |  |
| --- | --- | --- |
| Timestamp | Activity | Status |
| 06-10-2020-13:40 | Coding | "Output in single line" |
| 06-10-2020-13:45 | Coding | "floating value precision” |

1. data.txt

e 5 Elden

c 3 Felipe 55342.51415

e 3 Leonardo

e 1 Yong

c 9 Alessandra 8114.541862

c 6 Marnie 15287.78233

e 8 Kourtney

c 2 Lou 95053.44742

c 5 Numbers 51245.66138

e 4 Jarvis

e 9 Marlen

e 10 Florance

c 1 Devon 56442.27875

e 2 Elliott

c 8 Justina 73723.84849

c 10 Reyna 82946.53205

e 6 Antonetta

e 7 Florene

c 4 Merrill 98281.82784

c 7 Marlana 33252.21805

t 1 8 w 4924.86

t 9 6 d 3220.42

t 6 1 w 127.62

t 9 8 w 5566.7

t 9 8 d 5414.55

t 5 7 w 9422.35

t 9 10 d 1382.07

t 4 7 d 6131.07

t 8 2 w 2362.22

t 8 10 d 5834.48

t 5 4 w 5150.73

t 6 2 d 3795.96

t 1 9 w 3919.45

t 5 2 w 5037.31

t 3 9 w 8129.21

t 8 6 d 1235.67

t 2 4 d 6901.28

t 8 10 d 5599.44

t 6 9 d 1936.16

t 7 9 d 7363.98

1. testData.txt

123456789012345678901234567890123456789012345678901234567890

Devon Kourtney -$4924.86 $ 51517.42

Alessandra Antonetta +$3220.42 $ 11334.96

Marnie Yong -$127.62 $ 15160.16

Alessandra Kourtney -$5566.70 $ 5768.26

Alessandra Kourtney +$5414.55 $ 11182.81

Numbers Florene -$9422.35 $ 41823.31

Alessandra Florance +$1382.07 $ 12564.88

Merrill Florene +$6131.07 $ 104412.90

Justina Elliott -$2362.22 $ 71361.63

Justina Florance +$5834.48 $ 77196.11

Numbers Jarvis -$5150.73 $ 36672.58

Marnie Elliott +$3795.96 $ 18956.12

Devon Marlen -$3919.45 $ 47597.97

Numbers Elliott -$5037.31 $ 31635.27

Felipe Marlen -$8129.21 $ 47213.30

Justina Antonetta +$1235.67 $ 78431.78

Lou Jarvis +$6901.28 $ 101954.73

Justina Florance +$5599.44 $ 84031.22

Marnie Marlen +$1936.16 $ 20892.28

Marlana Marlen +$7363.98 $ 40616.20

1. Functions.py

# Variables

employeeDetail = dict()

customerDetail = dict()

transactionDetail = list()

transactionLog = list()

# Functions

def ReadDataOfFile(filename: str):

return open(filename, "r").readlines()

def CreateEmployeeDictionary(fileData: list):

for line in fileData:

line = line.strip().split()

if line[0] == "e":

id = line[1]

employeeDetail[id] = line[2]

def CreateCustomerDictionary(fileData: list):

for line in fileData:

line = line.strip().split()

if line[0] == "c":

id = line[1]

customerDetail[id] = dict()

customerDetail[id]["name"] = line[2]

customerDetail[id]["balance"] = float(line[3])

def CreateTransactionDictionary(fileData: list):

for line in fileData:

line = line.strip().split()

if line[0] == "t":

transactionData = dict()

transactionData["customerId"] = line[1]

transactionData["employeeId"] = line[2]

transactionData["type"] = line[3]

transactionData["amount"] = float(line[4])

transactionDetail.append(transactionData)

def Deposit(transaction: dict):

customerName = customerDetail[transaction["customerId"]]["name"]

employeeName = employeeDetail[transaction["employeeId"]]

customerDetail[transaction["customerId"]]["balance"] += transaction["amount"]

transactionLog.append(f"{ customerName } { employeeName } +${ transaction['amount'] :.2f } $ { customerDetail[transaction['customerId']]['balance'] :.2f }\n")

def Withdraw(transaction: dict):

customerName = customerDetail[transaction["customerId"]]["name"]

employeeName = employeeDetail[transaction["employeeId"]]

customerDetail[transaction["customerId"]]["balance"] -= transaction["amount"]

transactionLog.append(f"{ customerName } { employeeName } -${ transaction['amount'] :.2f } $ { customerDetail[transaction['customerId']]['balance'] :.2f }\n")

def Transaction(): transactionLog.append("123456789012345678901234567890123456789012345678901234567890\n")

for transaction in transactionDetail:

if transaction["type"] == "d":

Deposit(transaction)

else:

Withdraw(transaction)

def WriteTransactionLogToFile(filename: str):

with open(filename, "w") as fhandler:

fhandler.writelines(transactionLog)

print("Transaction log created !!!")

def TestResult(testData: list):

for i in range(len(transactionLog)):

if testData[i].strip() == transactionLog[i].strip():

print(f"Correct output for {i + 1}th data.")

else:

print(f"Incorrect output for {i + 1}th data.")

1. transactionLogGenerator.py

# Start point

# Functions

import Functions as fn

fileData = fn.ReadDataOfFile("data.txt")

fn.CreateEmployeeDictionary(fileData)

fn.CreateCustomerDictionary(fileData)

fn.CreateTransactionDictionary(fileData)

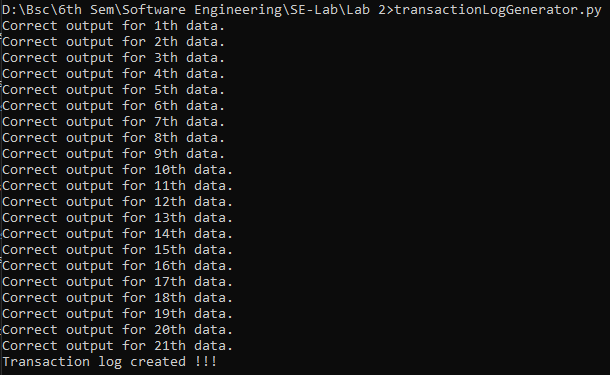
fn.Transaction()

testData = fn.ReadDataOfFile("testData.txt")

fn.TestResult(testData)

fn.WriteTransactionLogToFile("transactionLog.txt")

1. Output
   1. Command-line interface



* 1. transactionLog.txt

123456789012345678901234567890123456789012345678901234567890

Devon Kourtney -$4924.86 $ 51517.42

Alessandra Antonetta +$3220.42 $ 11334.96

Marnie Yong -$127.62 $ 15160.16

Alessandra Kourtney -$5566.70 $ 5768.26

Alessandra Kourtney +$5414.55 $ 11182.81

Numbers Florene -$9422.35 $ 41823.31

Alessandra Florance +$1382.07 $ 12564.88

Merrill Florene +$6131.07 $ 104412.90

Justina Elliott -$2362.22 $ 71361.63

Justina Florance +$5834.48 $ 77196.11

Numbers Jarvis -$5150.73 $ 36672.58

Marnie Elliott +$3795.96 $ 18956.12

Devon Marlen -$3919.45 $ 47597.97

Numbers Elliott -$5037.31 $ 31635.27

Felipe Marlen -$8129.21 $ 47213.30

Justina Antonetta +$1235.67 $ 78431.78

Lou Jarvis +$6901.28 $ 101954.73

Justina Florance +$5599.44 $ 84031.22

Marnie Marlen +$1936.16 $ 20892.28

Marlana Marlen +$7363.98 $ 40616.20