UEEN1043 Object-Oriented Concepts and Programming Techniques

Assignment 1

2025

Student Name	ID	Program
Teoh Zheng Wei	2400782	3E

```mermaid
flowchart TB
    %% Column 1 - Main Flow
    Start([Start])
    ReadT[Read Transactions]
    CleanT[Clean Transactions]
    ValidateT[Validate Transactions]
    ReformatD[Reformat Date]
    SortT[Sort Transactions]
    SaveT[Save Transactions]
    PrintO[/Print Output/]
    End([End])

    Start --> ReadT --> CleanT --> ValidateT --> ReformatD --> SortT --> SaveT --> PrintO --> End

    %% Column 2 - Read Transactions
    ReadTrans([Read Transactions])
    InitList[Initialize list "transactions"]
    OpenFile[Open file at "file_path" in read mode]
    CreateReader[Create "csv.DictReader" object]
    AppendRow[Append "row" to "transactions"]
    CloseFile[Close the file]
    ReturnList([Return "transactions" list])

    ReadTrans --> InitList --> OpenFile --> CreateReader --> AppendRow --> CloseFile --> ReturnList

    %% Column 3 - Clean Transactions
    CleanTrans([Clean Transactions])
    InitDict[Initialize dictionary "cleaned"]
    CheckEmpty{Check data are empty}
    CheckAmount{Check "amount" < 0}
    ConvertAmount[Convert "amount" to float and remove $]
    CreateKey[Create unique key]
    CheckDup{Check data duplicate}
    SkipTrans[Skip transaction]
    AddData[Add data to "cleaned"]
    Merge((  ))
    ReturnCleaned[Return "cleaned" as list]

    CleanTrans --> InitDict --> CheckEmpty
    CheckEmpty -->|No| CheckAmount
    CheckEmpty -->|Yes| SkipTrans
    CheckAmount -->|No| ConvertAmount
    CheckAmount -->|Yes| SkipTrans
    ConvertAmount --> CreateKey --> CheckDup
    CheckDup -->|Yes| SkipTrans
    CheckDup -->|No| AddData
    AddData --> Merge
    SkipTrans --> Merge
    Merge --> ReturnCleaned
```

```mermaid
flowchart TD
    A[Validate Transactions] --> B[Initialize list "valid_transactions"]
    B --> C{"use_chip" is "Swipe Transaction" or "Online Transaction"}
    C -->|Yes| D{"use_chip" is "Swipe Transaction" and "merchant_city" is "ONLINE"}
    C -->|Yes| E{"use_chip" is "Online Transaction" and "merchant_city" is not "ONLINE"}
    C -->|No| F[Skip transaction]
    D -->|Yes| F
    E -->|Yes| F
    D --> G((O))
    E --> G
    F --> G
    G --> H[Append "use_chip" to "valid_transactions"]
    H --> I[Return "valid_transactions" list]
```

Validate Transactions

Initialize list "valid_transactions"

"use_chip" is "Swipe Transaction" or "Online Transaction"

- Yes → **"use_chip" is "Swipe Transaction" and "merchant_city" is "ONLINE"**
- No → **Skip transaction**
- Yes → **"use_chip" is "Online Transaction" and "merchant_city" is not "ONLINE"**

"use_chip" is "Swipe Transaction" and "merchant_city" is "ONLINE" — Yes → **Skip transaction**

"use_chip" is "Online Transaction" and "merchant_city" is not "ONLINE" — Yes → **Skip transaction**

Append "use_chip" to "valid_transactions"

Return "valid_transactions" list

Reformat Date

Reformat Date

↓

Split date into
"date_parts[0]" (date)
and
"date_parts[1]" (time)

↓

Split "date_parts[0]"
into:
"y", "m", "d"

↓

Reformat date into "d"-
"m"-"y" "date_parts[1]"

↓

End

Sort Transactions

Sort Transactions

↓

Sort data by
"merchant_id", "date'
and "client_id" using
lambda key

↓

Return sorted list

Save Transactions

Save Transactions

↓

"Transactions"
empty —— Yes ——→ Return

↓ No

Open file in write mode

↓

Create "csv.DictWriter"
object

↓

Write the header row

↓

Add $ to amount

↓

Write the row to the file

↓

Close the file

↓

End