Universiti Tunku Abdul Rahman
UEEN1043 Object-Oriented Concepts and Programming Techniques
Assignment 1 (20%)
2025

Title: Transactions data management by applying Python containers and functions.

Problem Statement

Transactions data is provided in an input file named ***transactions.csv***. Each row in the file represents a single transaction record and includes the following details.

- **id** – Transaction identifier
- **date** – Transaction timestamp (YYYY-MM-DD HH:MM:SS format)
- **client_id** – Identifier for the client
- **card_id** – Identifier for the card used
- **amount** – Transaction amount
- **use_chip** – Type of transaction ("Swipe Transaction" or "Online Transaction")
- **merchant_id** – Identifier for the merchant
- **merchant_city** – City where the transaction occurred
- **errors** *(optional)* – Error details, if any

Your program should perform the following tasks. You may use your creativity in designing your program to fulfil the specification stated according to the marking rubric.

1. Data cleaning

   - Remove any transactions where **date**, **client_id**, **card_id**, or **amount** is missing.
   - Drop duplicate transactions where (**date, client_id, card_id, amount**) are identical, retaining only the most recent (last) record.
   - Keeps only transactions where **amount** is not null.
   - Transactions with a negative **amount** are invalid and should be removed (zero is allowed).

2. Validate transactions

   - The **use_chip** field must be either "Swipe Transaction" or "Online Transaction".
   - Ensure that **use_chip** is "Online Transaction" only if **merchant_city** is "ONLINE", and vice versa.
   - Remove transactions that fail these conditions.

3. Reformat date

   - Change the format from YYYY-MM-DD HH:MM:SS to DD-MM-YYYY HH:MM:SS.

4. Sort and save the processed data

   - Sort the transactions by **merchant_id**, **date**, and **client_id** (in ascending order).
   - Save the final processed data as ***processed_transactions.csv***.

Assessment (Individual)

- Your solution should make use of the container classes in Python, e.g. list and dict. You can only use the Python built-in functions to perform the task. Use of any advanced modules other than csv, such as pandas and numpy is not allowed.
- Your submission should include the following in a zip file:
  - An executable Python program (.py) with clear and concise comments. State your program, name, id in the .py file.
  - An output file ***processed_transactions.csv***
  - A brief report (.pdf) which includes the program flowchart – concise, informative, not too lengthy. Use the cover page provided to prepare your report.
- Name your zip file according to the program, student's name, and ID, e.g. <program>_<name>_<id>.zip. For example, 3E_ChuaSingYee_20250101.zip.

Deadline: **5 pm, 7 April 2025 (Week 9 Monday).**
Marks will be deducted for late submission.

Warning!!! Plagiarism will result in failure (high similarity for program or flowchart).

Marking Rubric

Criteria	0	1	2	3	4	5
Flowchart clearly explains the solution (5)						
Modular programming effectively implements the flowchart (5)						
Program correctness & robustness (5)						
Data cleaning & transactions validation (5)						
Data sorting & formatting (5)						
Efficient use of Python containers, optimized performance, avoid 3 or more levels of nested loops (5)						
Systematic & reusability (5)						
Code readability & clarity (5)						
Total (/40)						
Contribution (/20%)						