CS 353 – Functional programming
Spring 2024
Program 3

For this program, we're going to be working with tagged data to model a payment system. The amount of information we'll track is much less than would be needed for an actual system, but is enough to give you some useful practice.

An account consists of a list of data:

- Account number (integer)
- Customer information (for our purposes, a string for the name)
- A current balance (float, default 0.0)

Account information will be provided in a text file, one account per line. For this assignment, the account list is static; there will be no accounts added or deleted by the program.

A purchase consists of:

- The tag 'purchase
- The account number of the purchaser
- A timestamp (integer)
- A merchant (string)
- An amount (float)

A payment consists of:

- The tag 'payment
- The account number this payment is for
- A timestamp (integer)
- A payment may be cash, check, or card.
- A cash payment consists of:
  - The tag 'cash
  - An amount > 0 (float)
- A check payment consists of:
  - The tag 'check
  - The check number (integer)
  - An amount > 0 (float)
- A card payment may be either credit or debit. (We need to distinguish because the two cards have different fee structures.)
  - A credit card payment consists of:
    - The tag 'credit
    - The card number (large integer, can treat as a string; we won't be doing calculations with it)
    - An amount > 0 (float)
  - A debit card payment consits of:
    - The tag 'debit
    - The card number (large integer, can treat as a string; we won't be doing calculations with it)
    - An amount > 0 (float)

You are given 2 data files: ACCOUNTS.TXT & TRANSACTIONS.TXT. The first is a list of accounts, as specified above. The second is a list of transactions. Tags will be unquoted strings. Names will be quoted strings, to allow whitespace. Each transaction will be on a separate line and end with a newline.

Your program should read the accounts file as part of initial setup, then process the transactions as follows:

- Each transaction is assigned a number. The first transaction is # 10001. Transactions are numbered sequentially in the order processed.
- Each transaction must be assigned to the correct account. How you do that is up to you (see 'Programming Notes,' below.)

Your program is to produce an output file, STATEMENTS.TXT, listing the current status of each account.

- Statements are sorted by account number.
- Each statement begins with a header, consisting of the account number and name, followed by the starting balance and a newline.
- The header is followed by the transactions, in timestamp order. Each transaction is on a separate line, formatted in columns.
    - Every transaction begins with the timestamp.
    - Purchases have the following, in columns (text columns left-aligned, numeric amounts right-aligned):
        - The word "Purchase"
        - The name of the merchant. (If the merchant's name is long,the first 40 characters of the merchant's name.)
        - The amount of the purchase
    - Payments have the following, in columns (similar to purchases):
        - The word "Payment"
        - The form of payment: Exactly one of "Cash", "Check", "Credit", "Debit". It is not necessary to list the check number or card number.
        - The amount of the payment.
- After the transaction list is the following; again, numeric values are right-aligned in columns, for legibility.
    - Total purchases
    - Total payments
    - Ending balance (starting balance + total purchases – total payments).

This should be followed by a separator to clearly mark the end of one statement and the beginning of the next.

Programming Notes:

- The data will be small enough to fit in memory. Think carefully about how to proceed, given that data is immutable. (Hint: Filtering or sorting data by account number would be a good first step.)
- Assume that the input files are in the same folder as your source code. You can hard-code the file names into the program. Don't ask the user for a path; likewise, don't put them into a subfolder or hard-code the entire path—I don't have access to the C: drive on your laptop.
- All account numbers will be valid—that is, there will be an account corresponding to the account on every transaction. For card transactions, the number won't necessarily pass validation as a valid CC number.

Upload your source file or the link to your online repository by the deadline.