Java Development Fundamentals

Capstone 1:
Accounting Ledger Application

Java CLI Application

Capstone Setup

You will create three capstones during this academy. They will be great projects to add to your portfolio to demonstrate to any potential employer or hiring manager.

We suggest that you create a new folder called `LearnToCode_Capstones` in your `C:/pluralsight` directory and keep local copies of your capstone projects there. In addition, we suggest you use a great name for the project in GitHub as employers may look at your repos during the hiring process.

You should create a GitHub project board to manage your work. Use the project requirements to create user stories on your board.

Description

In this project, you will use what you have learned about Java programming to create a CLI application. With this application you can track all financial transactions for a business or for personal use.

All transactions in the application should be read from and saved to a transaction file named `transactions.csv`. Each transaction should be saved as a single line with the following format.

```
date|time|description|vendor|amount
2023-04-15|10:13:25|ergonomic keyboard|Amazon|-89.50
2023-04-15|11:15:00|Invoice 1001 paid|Joe|1500.00
```

Application Requirements

Your application must include several screens with the listed features in order to be considered complete:

- **Home Screen**
  - The home screen should give the user the following options. The application should continue to run until the user chooses to exit.
    - **D) Add Deposit** - prompt user for the deposit information and save it to the csv file
    - **P) Make Payment (Debit)** - prompt user for the debit information and save it to the csv file
    - **L) Ledger** - display the ledger screen
    - **X) Exit** - exit the application
- **Ledger** - All entries should show the newest entries first
  - **A) All** - Display all entries
  - **D) Deposits** - Display only the entries that are deposits into the account
  - **P) Payments** - Display only the negative entries (or payments)
  - **R) Reports** - A new screen that allows the user to run pre-defined reports or to run a custom search
    - **1) Month To Date**
    - **2) Previous Month**
    - **3) Year To Date**
    - **4) Previous Year**
    - **5) Search by Vendor** - prompt the user for the vendor name and display all entries for that vendor
    - **0) Back** - go back to the report page
  - **H) Home** - go back to the home page

Challenge Yourself

If you have time and want to challenge yourself, consider the following:

On the reports screen add another option for a custom search. Prompt the user for search values for all ledger entry properties.

- **6) Custom Search** - prompt the user for the following search values.
    - Start Date
    - End Date
    - Description
    - Vendor
    - Amount
- If the user enters a value for a field you should filter on that field
- If the user does not enter a value, you should not filter on that field

Other Requirements

Your project must also meet the following requirements:

Repository

- Your code must be in a **public** GitHub repository
- The repository must contain an appropriate Git commit history
    - At a **minimum**, you should have a commit for each meaningful piece of work completed
- It must contain an informative **README** file that:
    - Describes your project
    - Includes images of your application screens
    - Describes/shows one interesting piece of code from your project

Class Demonstrations

Each student will be given 10 minutes to demonstrate their project to the class on "project demonstration day". During this time, you will:

- Present your application - run through the different screens and scenarios
- Describe / show one interesting piece of code that you wrote
- Answer questions from the audience if time permits