MIDTERM PROJECT

ASSIGNED Thursday 2/8/2018; DUE Monday 2/12/2018, 1:00 PM

The Midterm Project is an opportunity for bootcamp students to

- Begin to pull together a number of different things they've learned so far, including classes and polymorphism, user input, arrays and collections, file I/O, and exceptions;
- Work on a larger and more complicated case study than lab exercises as a prelude to the Final Project;
- Collaborate with teammates on a software project, experiencing the need for proper object-oriented design, documentation, and version control.

Every student will participate in this project in an assigned group. The majority of 3 days will be committed to the project (although outside work may still be necessary) and there will be multiple check-ins during those days to make sure groups are on-track.

The entire group will work together on one of the three projects. Take a little time to decide which one you want to tackle, but once you get started stick to that project—no turning back!

Possible projects:

- Minefield (a text-based version of Windows Minesweeper)
- Point-of-Sale Terminal (a cash register/ordering terminal for someplace like a store, coffee shop, or fast-food restaurant)
- Library System

See the following pages for more information on each project. Please recognize that the descriptions are minimum versions; it's hoped each group will go beyond these requirements and incorporate features of interest to them.

MINEFIELD

(See http://minesweeperonline.com/ if you aren't familiar with this game.)

Write a console Minefield application. At a minimum, this game should include:

- A class to hold the minefield in a 2D array, with methods to generate the minefield and to check a specific cell.
    - You are responsible for generating not just the mines but all the numbers in cells with adjacent mines!!
- A way of displaying the current state of the minefield (with symbols for unknown, empty, and flagged mines). There are several ways to do this either in the main app or in the minefield; some are better than others.
- A main class which takes input from the user:
    - Ask for the size of the minefield (either provide a menu of at least 3 sizes you've specified or allow any custom size up to a maximum number of rows/columns). The number of mines could either be set by the size or entered by the user.
    - Ask what they want to do next, flag a mine or uncover a cell.
        - If they uncover a mine, display a game over screen with the full solution of the minefield.
        - If they uncover an empty cell, uncover all the empty cells and numbers adjacent (like the sample online minesweeper) and re-display the board. (Ideal but optional)
        - If they uncover a cell with a number, just uncover that cell and re-display the board.
        - If they flag the last mine, display a win screen with the full solution.

Possible enhancements:

- (Easy/Medium) Allow the user a third option: Flag a cell with a question mark.
- (Hard) Incorporate graphics. You can make it a Windows Desktop Application.

POS TERMINAL

(That stands for Point-of-Sale, but what you think of your project is up to you.)

Write a cash register or self-service terminal for some kind of retail location. Obvious choices include a small store, a coffee shop, or a fast food restaurant.

- Your solution must include some kind of a product class with a name, category, description, and price for each item.
- 12 items minimum; they must be stored in a text file your program reads in.
- Present a menu to the user and let them choose an item (by number or letter).
    - Allow the user to choose a quantity for the item ordered.
    - Give the user a line total (item price * quantity).
- Either through the menu or a separate question, allow them to re-display the menu and to complete the purchase.
- Give the subtotal, sales tax, and grand total.
- Ask for payment type—cash, credit, or check
- For cash, ask for amount tendered and provide change.
- For check, get the check number.
- For credit, get the credit card number, expiration, and CVV.
- At the end, display a receipt with all items ordered, subtotal, grand total, and appropriate payment info.
- Return to the original menu for a new order. (Hint: you'll want an array or ArrayList to keep track of what's been ordered!)

Optional enhancements:

- (Moderate) Include an option to add to the product list, which then outputs to the product file.
- (Hard) Create a full GUI.

LIBRARY TERMINAL

Write a console program which allows a user to search a library catalog and reserve books.

- Your solution must include some kind of a book class with a title, author, status, and due date if checked out.
    - Status should be On Shelf or Checked Out (or other statuses you can imagine). For additional practice, use an Enum if you want!
- 12 items minimum; they must be stored in a text file your program reads in.
- Allow the user to:
    - Display the entire list of books. Format it nicely.
    - Search for a book by author.
    - Search for a book by title keyword.
    - Select a book from the list to check out.
        - If it's already checked out, let them know.
        - If not, check it out to them and set the due date to 2 weeks from today.
    - Return a book. (You can decide how that looks/what questions it asks.)
- When the user quits, save the current library book list (including due dates and statuses) to the text file so the next time the program runs, it remembers.

Optional enhancements:

- (Moderate) Include an option to add to the book list, which then outputs to the book file.
- (Hard) Create a full GUI.