CSE 1322L – Assignment 1 (Spring 2025)

Introduction

In this assignment, you will write a simple program to help a small store keep track of its inventory, of which the store only has two: apples and oranges. The program needs to keep track of how much of each product is in store, how much either product is currently being sold for, as well as the store's balance. While the store cannot store fractions of fruits (only whole fruits), the store is allowed to go infinitely into debt.

Requirements

The features described below must be in your program:

- Your main method must keep track of the following, using appropriate data types:
  - Number of apples in storage, starting at 0
  - Number of oranges in storage, starting at 0
  - The current sell price of apples, starting at 0
  - The current sell price of oranges, starting at 0
  - The store's current balance, starting at 0
- Your main method must implement the following menu options:
  - **Buy apples**
    - Prompts the user for the number of apples to buy.
    - Do not accept negative values: keep prompting the user for the number of apples to buy until they enter a non-negative value.
    - If the user enters "0", return them to the main menu
    - For any positive number, ask the user the price at which those apples are being bought for. You don't need to check for negative values here, as the store could potentially be paid to take in inventory. Finally print "Bought X apples at $Y for a total of $Z", where X is the number of apples bought, Y is the amount each apple was bought for, and Z is the total amount the apples were bought for
    - Be sure to accurately track the changes in the store's number of apples and balance. The store's balance can become negative, but its inventory cannot
  - **Buy oranges**: Same as above, but for oranges
  - **Sell apples**:
    - Prompts the user for the number of apples to sell
    - Do not allow the user to sell a negative number of apples. Keep prompting them for the number of apples to sell until they enter a non-negative number

        - If the user tries to sell less apples than the store has, print "Sold X apples @ $Y for a total of $Z", where X is the number of apples sold, Y is the amount each apple was sold for, and Z is the total amount the apples were sold for
        - If the user tries to sell exactly as many apples as the store has, print "Sold **all** apples @ $Y for a total of $Z". Same variables apply as above
        - If the user tries to sell more apples than the store has, print "**Not enough apples**. Selling instead X apples @ $Y for a total of $Z", where X is instead the current number of apples the store has, and Y and Z are the same as above
        - Be sure to accurately track the changes in the store's number of apples and balance.
    - **Sell oranges**: Same as above, but for oranges
    - **Change price of apples**: Prompts the user for a new selling price for the apples
        - Do not allow negative values. If the user tries to enter one, keep prompting them until they enter a non-negative value
        - A value of 0 is acceptable
    - **Change price of oranges**: Same as above, but for oranges
    - **List inventory**: Prints the following string:

      ```
      "Current inventory is:
      X apples currently selling @ $Y
      Z oranges currently selling @ $W"
      ```

      Where X is the current number of apples in storage, Y is the apple's current sell price, Z is the current number of oranges in storage, and W is the orange's current sell price
    - **Show balance**: Prints the store's current balance
    - **Quit**: Terminates the program

Deliverables

- Assignment1.java (driver)

Considerations

- You will get partial credit for partial work, as long as the rubric permits it. If you yourself near the deadline and it's too late to ask for help, try to do as much as you can.
- For variables of type double, you will **not** lose points due to rounding errors. If the sample output below says "1.00001" and yours says "1.00002", you will still get full

credit **as long as your logic is correct.** Similarly, you do **not** need to round off your numbers to a specific number of decimal places. If the sample output says "2.75" and yours says "2.747155482733", you will still get full credit **as long as your logic is correct.**

- Be mindful of the data types you will use for each variable. Ask yourself what values are acceptable for that variable to hold, and that will usually tell you what data type should be used.
- Remember that, unless stated otherwise, you should prefer to use "int" for data which will hold whole numbers and "double" for data that will hold numbers with a decimal component.
- While you can write any loop as either a FOR loop, a WHILE loop, or a DO-WHILE loop, choosing the correct loop may help communicate to whoever is reading your code what that loop is intended to be used for. Same applies between IF blocks and SWITCH statements.
- You will notice that some of the options are nearly identical, save for what variables they are handling and specific parts of their string literals
    - It might be best to write the option for apples first and, once you've verified that it works correctly, copy and paste it onto the option for oranges, replacing as appropriate
    - Be extra careful when performing a replacement. Use the text finder (ctrl + f on Windows, command + f on Mac) to help you locate all instances of a word in your code
    - In a future module, we will see some techniques to avoid repetitive code

Sample Output (user input in red)

```
[Fruit Management System]
1. Buy apples
2. Buy oranges
3. Sell apples
4. Sell oranges
5. Change price of apples
6. Change price of oranges
7. List inventory
8. Show balance
0. Quit
Enter option: 7
Current inventory is:
0 apples currently selling @ $0.00
0 oranges currently selling @ $0.00

1. Buy apples
2. Buy oranges
```

```
3. Sell apples
4. Sell oranges
5. Change price of apples
6. Change price of oranges
7. List inventory
8. Show balance
0. Quit
Enter option: 8
Current balance is $0.00

1. Buy apples
2. Buy oranges
3. Sell apples
4. Sell oranges
5. Change price of apples
6. Change price of oranges
7. List inventory
8. Show balance
0. Quit
Enter option: 1
Buy how many apples? 100
Buy apples at what price? $2.10
Bought 100 apples @ $2.10 for a total of $210.00

1. Buy apples
2. Buy oranges
3. Sell apples
4. Sell oranges
5. Change price of apples
6. Change price of oranges
7. List inventory
8. Show balance
0. Quit
Enter option: 2
Buy how many oranges? -10
Invalid quantity. Enter a non-negative number: 0
Buying no oranges.

1. Buy apples
2. Buy oranges
3. Sell apples
4. Sell oranges
```

```
5. Change price of apples
6. Change price of oranges
7. List inventory
8. Show balance
0. Quit
Enter option: 2
Buy how many oranges? 10
Buy oranges at what price? $3.25
Bought 10 oranges @ $3.25 for a total of $32.50

1. Buy apples
2. Buy oranges
3. Sell apples
4. Sell oranges
5. Change price of apples
6. Change price of oranges
7. List inventory
8. Show balance
0. Quit
Enter option: 5
What should be the new price of selling apples? $5.20
Selling price of apples set @ $5.20

1. Buy apples
2. Buy oranges
3. Sell apples
4. Sell oranges
5. Change price of apples
6. Change price of oranges
7. List inventory
8. Show balance
0. Quit
Enter option: 6
What should be the new price of selling oranges? $-7.50
Invalid price. Enter a non-negative price: $7.50
Selling price of oranges set @ $7.50

1. Buy apples
2. Buy oranges
3. Sell apples
4. Sell oranges
5. Change price of apples
```

```
6. Change price of oranges
7. List inventory
8. Show balance
0. Quit
Enter option: 7
Current inventory is:
100 apples currently selling @ $5.20
10 oranges currently selling @ $7.50

1. Buy apples
2. Buy oranges
3. Sell apples
4. Sell oranges
5. Change price of apples
6. Change price of oranges
7. List inventory
8. Show balance
0. Quit
Enter option: 3
Sell how many apples @ $5.20? 50
Sold 50 apples @ $5.20 for a total of $260.00

1. Buy apples
2. Buy oranges
3. Sell apples
4. Sell oranges
5. Change price of apples
6. Change price of oranges
7. List inventory
8. Show balance
0. Quit
Enter option: 4
Sell how many oranges @ $7.50? -10
Enter a non-negative number of oranges to sell: -10
Enter a non-negative number of oranges to sell: 10
Sold all oranges @ $7.50 for a total of $75.00

1. Buy apples
2. Buy oranges
3. Sell apples
4. Sell oranges
5. Change price of apples
```

```
6. Change price of oranges
7. List inventory
8. Show balance
0. Quit
Enter option: 5
What should be the new price of selling apples? $6.05
Selling price of apples set @ $6.05

1. Buy apples
2. Buy oranges
3. Sell apples
4. Sell oranges
5. Change price of apples
6. Change price of oranges
7. List inventory
8. Show balance
0. Quit
Enter option: 3
Sell how many apples @ $6.05? 60
Not enough apples. Selling instead 50 apples @ $6.05 for a total of
$302.50

1. Buy apples
2. Buy oranges
3. Sell apples
4. Sell oranges
5. Change price of apples
6. Change price of oranges
7. List inventory
8. Show balance
0. Quit
Enter option: 7
Current inventory is:
0 apples currently selling @ $6.05
0 oranges currently selling @ $7.50

1. Buy apples
2. Buy oranges
3. Sell apples
4. Sell oranges
5. Change price of apples
6. Change price of oranges
```

```
7. List inventory
8. Show balance
0. Quit
Enter option: 8
Current balance is $395.00

1. Buy apples
2. Buy oranges
3. Sell apples
4. Sell oranges
5. Change price of apples
6. Change price of oranges
7. List inventory
8. Show balance
0. Quit
Enter option: 0
Shutting off...
```