CIS355A Week 5 Lab—File Processing

**Objectives**

* Add persistent data storage to your Week 4 Lab using text file input and output.

**Problem: Stocks4U Portfolio Management System**

The portfolio management system you developed for Stocks4U needs the ability to save and restore a user’s data from a text file.

**Functional Requirements**

You can code the GUI by hand or use NetBeans GUI builder interface.

You will enhance Week 4 GUI to include the following.

* A file menu with menu items: open, save, and exit
* A label to display total portfolio value

Stock Class

* Modify the toString of Stock class to display as

“Company: qty shares” (i.e., “Apple: 10 shares”)

StockIO Class

Create a StockIO class that is used to read from and write to a text file using an ArrayList. Make sure to use a delimiter between the fields; it does not have to be the # character. Example format of the file is as follows.

Apple#100#55.0#80.0

Intel#50#75.0#70.0

This class should have two methods.

* getData—reads data from file, returns data in array list of stock objects
* saveData—writes data from an array list to the file in proper format

The file name will be an instance variable that you can set with a parameterized constructor, or with a separate method.

GUI Class

Note that you will need to add an ArrayList to your GUI class to manage the data to and from the file. It will act as a parallel array to your DefaultListModel. Any time you add a stock, you must add it in both places. Any time you remove a stock, you must remove it in both places.

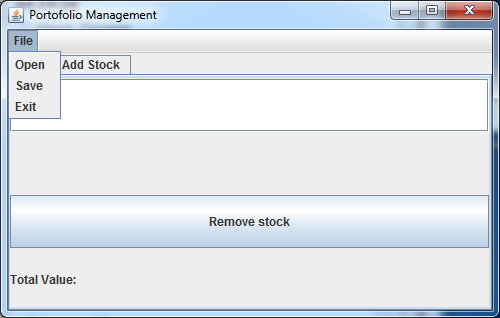
File—Open should prompt for file name using JOptionPane, read the file and populate the JList.

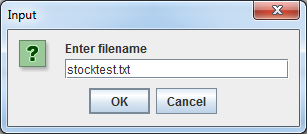
File—Save should prompt for file name to save data from JList to.

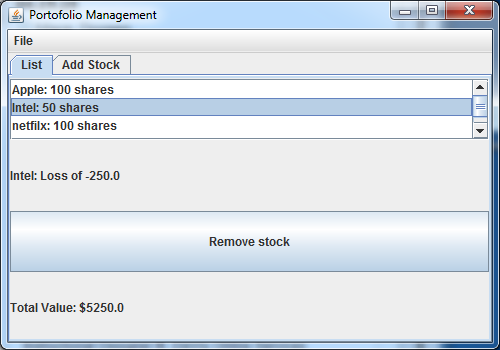
File—Exit should exit the program.

The total value of the portfolio should always be displayed and updated anytime a stock is added or removed.

**Sample GUI**







**RUBRIC**

|  |  |
| --- | --- |
| Stock class toString modified | 5 |
| GUI class   * Menu is added. * Label is added for total portfolio value. * Open menu item reads data from a file and displays in list box. * Save menu item writes data to a text file in proper format. * Total value is updated whenever any changes are made to stocks (add, remove, open a new file). | 15 |
| StockIO class   * getData method reads from file to array list. * saveData method writes from array list to file. | 15 |
| Code style | 5 |
| Total | 40 |

**Code Style Requirements**

* Include meaningful comments throughout your code.
* Use meaningful names for variables.
* Code must be properly indented.
* Include a comment header at beginning of each file (example below).

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
Program Name: ProgramName.java   
Programmer's Name: Student Name   
Program Description: Describe here what this program will do   
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**Deliverables**

Submit the following as a single zip folder.

* All Java files

Follow assignment specification regarding class and method names.

Note that your Java file name must match class name (do not rename).