



**Objective:** The objective of this project is to practice using files of objects and gain more experiences in developing interactive GUI using Java Swing and AWT toolkits.

**Project description:**

In this assignment, you are to design a Java program to record checking account transactions and to keep track of the amount of money in your checking account. A transaction record should include the following information:

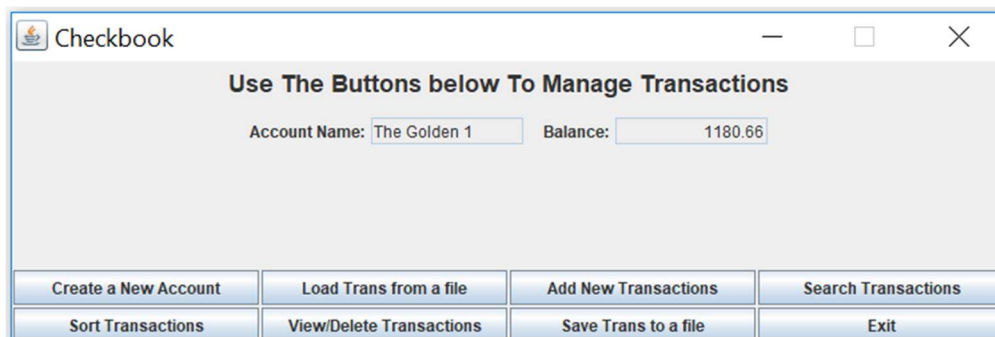
- . Transaction date, **a string of characters**
- . Transaction type; Deposit, Automatic deposit, ATM Withdrawal, Check, or Debit Card.
- . Check No., **an integer**
- . Transaction description, **a string of characters**
- . Amount, **a floating point number**

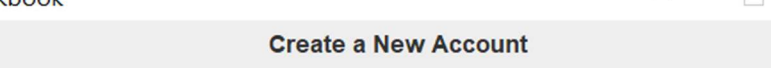
Your program should allow an user to do the following:

1. Create a new account
2. Load transactions from a file
3. Add new transactions
4. Search transactions (optional)
5. Sort transactions (optional)
6. View/Delete transactions
7. Save transactions to a file

**User interface requirement:**

Your program should have a GUI closely similar to the following.





The screenshot shows a window titled "Checkbook" with a standard macOS title bar (red, yellow, and green buttons). The window content is a light gray dialog box titled "Create a New Account" in bold black text. Below the title, there are two text input fields. The first is labeled "Account Name:" and the second is labeled "Initial Balance:". Both labels are in a standard black font. At the bottom of the dialog box, there are two buttons: "Create" and "Cancel". Both buttons have a light blue gradient and a thin black border.

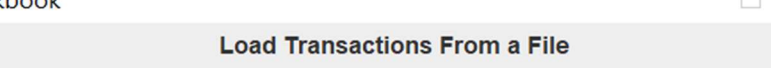
Checkbook

**Create a New Account**

Account Name:

Initial Balance:

Create Cancel



Checkbook

— □ ×


**Load Transactions From a File**

Account Name:

The screenshot shows the 'Checkbook' application window. The title bar includes a logo and the text 'Checkbook'. The menu bar contains 'File', 'Edit', 'View', and 'Help'. The main form area has the following fields:

- Date: [Text Field]
- Trans. Type: [Dropdown Menu]
- Check No.: [Text Field]
- Trans. Description: [Text Field]
- Payment/Debit(-): [Text Field]
- Deposit/Credit(+): [Text Field]

At the bottom of the window are two buttons: 'Save New Transaction' and 'Top Menu'.


Checkbook
—
□
✕

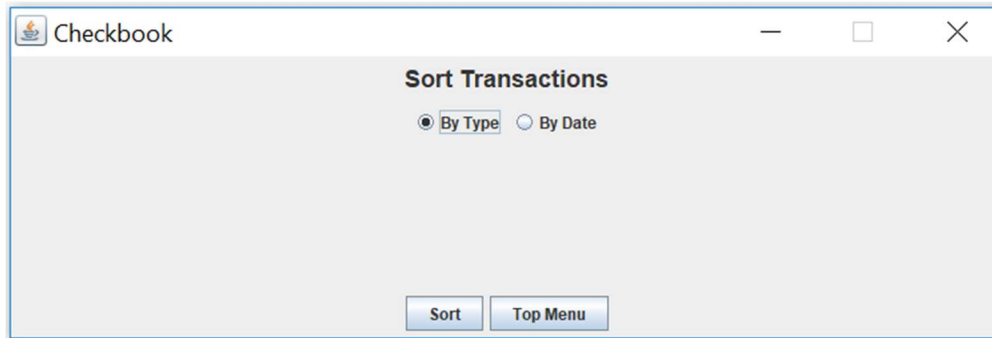
Search Transactions by Transaction Date/Type/Check No./Description

Date	Trans. Type	Check No.	Trans. Description	Payment/Debit(-)	Deposit/Credit(+)	Balance

Search String:

Search

Top Menu



Transactions Currently In The Checkbook						
Date	Trans. Type	Check No.	Trans. Description	Payment/Debit(-)	Deposit/Credit(+)	Balance
01/01/2011	Automatic ...		Paycheck		1350.00	1450.00
01/02/2011	Check	100	Steakhouse	35.75		1414.25
01/02/2011	ATM Withdr...			18.58		1395.67
01/05/2011	Check	102	Dollar Store	17.32		1378.35
01/11/2011	Debit Card		Food Mart	100.00		1278.35
01/20/2011	Check	103	Electric Bill	75.35		1203.00
02/02/2011	Check	105	PG&E	22.34		1180.66

### To submit your project:

1. A hardcopy of your Java source code.
2. Email a softcopy of your source code.
3. Demo your program on athena before **12/07/2017**.

### Programming hints about JLabel, JTable and JScrollPane:

1. To change the font size of a JLabel.

```
new JLabel("<html><font size=5><b>JLabel text</b></html>");
```

2. Create a JScrollPane and add it to the center of the contentPane of a frame.

```
scrollPane = new JScrollPane();
contentPane.add(scrollPane, BorderLayout.CENTER);
```

3. Before display contacts.

- a. Create a JTable.

```
JTable abtable = new JTable(data, columnName);
```

- b. Use the JTable to create a temporary JScrollPane.

```
JScrollPane tmp = new JScrollPane(abtable);
```

- c. Get the viewport from the temporary JScrollPane and add it to the original JScrollPane.

```
scrollPane.setViewport(tmp.getViewport());
```