

# CS1083 Assignment # 5 - Fall 2024

---

**Due: Wednesday, 16 October before 4:30 pm in the Desire2Learn dropbox. (See submission instructions below).**

The purpose of this assignment is:

- Handle exceptions in a textual user interface.
- Gain experience writing your own exceptions.

**This assignment is to be done individually. If you have questions, direct them to a tutor/assistant during an extra help session. If your question is not answered during an extra help session, you may contact your course professor.**

## Background Scenario

Your task is to create a basic textual user interface demo for a bank that allows a user to keep track of their various bank accounts. It is important that this program does not terminate running due to exceptions.

## Account Types

There are two types of accounts that a user can have: (1) a chequing account, and (2) a savings account. A user may have multiple of each. Both accounts should be objects that have a constructor that takes a parameter amount and stores it as the starting amount, an accessor and mutator for the amount, an apply interest method, and withdraw/deposit methods. Chequing and savings accounts are mostly the same except for how they calculate interest. Chequing accounts will always have a flat interest amount of 0.005% while savings accounts will take an additional parameter as a constructor being the interest amount. If you are not sure how to apply interest, you may wish to do some research. **Keep in mind good programming principles here, especially code reusability.**

## Account Exceptions

You will need to write two new types of exceptions:

(1) InsufficientFundsException which will be thrown if:

- a. A withdraw is being made for more money than what is stored in the account. The exception should have a message detailing how much extra beyond the amount in the account is trying to be withdrawn.

(2) NegativeException which will be thrown if:

- a. A deposit or withdraw is requested with a negative amount.
  - b. When constructing a savings account, the interest amount is negative.
- All exceptions should be thrown with a message that specifically states what the issue was. For example “**Deposit** amounts cannot be negative.”

## User Interface

Finally, write a textual user interface with the following options:

User Input	Execution
1	Create a new Chequing Account (start with a balance of 0)
2	Create a new Savings Account (start with a balance of 0). Will need to ask the user for the percentage of interest. Accounts with improper interest values should not be registered.
3	Deposit to current account. Will need to ask user for an amount.
4	Withdraw from current account. Will need to ask user for an amount.
5	Apply interest to the current account.
6	Allow the user to switch to a different account based on ID. Will need to ask the user for this. If account does not exist, print message, and ask for another command.
0	Stop execution.

See example command loop below.

Upon each command loop, the current account number and balance should be displayed. If no accounts exist, do not allow the user to select any other input beyond 1, 2, or 0.

If the user gives textual input at any point as opposed to numeric input, catch the InputMismatchException, clear the scanner, and state that there was an error with the input.

## Example Output

See some example output below (user input is highlighted):

### Example 1: Options 1, 2, 3, 5, 6

NO ACCOUNTS MADE. PLEASE CREATE AN ACCOUNT.

Please input a command:

- 1: Create a new Chequing Account
- 2: Create a new Savings Account
- 3: Deposit Funds
- 4: Withdraw Funds
- 5: Apply Interest
- 6: Switch to a Different Account
- 0: To Exit

4

NO ACCOUNTS MADE. PLEASE CREATE AN ACCOUNT.

Please input a command:

- 1: Create a new Chequing Account
- 2: Create a new Savings Account
- 3: Deposit Funds
- 4: Withdraw Funds

5: Apply Interest  
6: Switch to a Different Account  
0: To Exit

1

Currently working with account: 0 Balance of \$0.0

Please input a command:

1: Create a new Chequing Account  
2: Create a new Savings Account  
3: Deposit Funds  
4: Withdraw Funds  
5: Apply Interest  
6: Switch to a Different Account  
0: To Exit

2

Input a non-negative interest amount:

0.12

Currently working with account: 1 Balance of \$0.0

Please input a command:

1: Create a new Chequing Account  
2: Create a new Savings Account  
3: Deposit Funds  
4: Withdraw Funds  
5: Apply Interest  
6: Switch to a Different Account  
0: To Exit

3

Input an amount to deposit:

-100

Deposit amounts must be positive.

Currently working with account: 1 Balance of \$0.0

Please input a command:

1: Create a new Chequing Account  
2: Create a new Savings Account  
3: Deposit Funds  
4: Withdraw Funds  
5: Apply Interest  
6: Switch to a Different Account  
0: To Exit

3

Input an amount to deposit

4000

Currently working with account: 1 Balance of \$4000.0

Please input a command:

1: Create a new Chequing Account  
2: Create a new Savings Account  
3: Deposit Funds  
4: Withdraw Funds  
5: Apply Interest  
6: Switch to a Different Account  
0: To Exit

5

Currently working with account: 1 Balance of \$4480.0

Please input a command:

1: Create a new Chequing Account  
2: Create a new Savings Account  
3: Deposit Funds  
4: Withdraw Funds  
5: Apply Interest  
6: Switch to a Different Account  
0: To Exit

6

Which account would you like to switch to?

0

Currently working with account: 0 Balance of \$0.0

Please input a command:

- 1: Create a new Chequing Account
- 2: Create a new Savings Account
- 3: Deposit Funds
- 4: Withdraw Funds
- 5: Apply Interest
- 6: Switch to a Different Account
- 0: To Exit

0

## Example 2: Options 2, 3, 4, 6

NO ACCOUNTS MADE. PLEASE CREATE AN ACCOUNT.

Please input a command:

- 1: Create a new Chequing Account
- 2: Create a new Savings Account
- 3: Deposit Funds
- 4: Withdraw Funds
- 5: Apply Interest
- 6: Switch to a Different Account
- 0: To Exit

2

Input a non-negative interest amount:

-10

Interest rates may not be negative.

NO ACCOUNTS MADE. PLEASE CREATE AN ACCOUNT.

Please input a command:

- 1: Create a new Chequing Account
- 2: Create a new Savings Account
- 3: Deposit Funds
- 4: Withdraw Funds
- 5: Apply Interest
- 6: Switch to a Different Account
- 0: To Exit

2

Input a non-negative interest amount:

0.15

Currently working with account: 0 Balance of \$0.0

Please input a command:

- 1: Create a new Chequing Account
- 2: Create a new Savings Account
- 3: Deposit Funds
- 4: Withdraw Funds
- 5: Apply Interest
- 6: Switch to a Different Account
- 0: To Exit

3

Input an amount to deposit

5000

Currently working with account: 0 Balance of \$5000.0

Please input a command:

- 1: Create a new Chequing Account
- 2: Create a new Savings Account
- 3: Deposit Funds
- 4: Withdraw Funds
- 5: Apply Interest
- 6: Switch to a Different Account
- 0: To Exit

4

Input an amount to withdraw

300

Currently working with account: 0 Balance of \$4700.0

Please input a command:

- 1: Create a new Chequing Account
- 2: Create a new Savings Account

3: Deposit Funds  
4: Withdraw Funds  
5: Apply Interest  
6: Switch to a Different Account  
0: To Exit

4

Input an amount to withdraw

-300

Withdrawal amounts must be positive

Currently working with account: 0 Balance of \$4700.0

Please input a command:

1: Create a new Chequing Account  
2: Create a new Savings Account  
3: Deposit Funds  
4: Withdraw Funds  
5: Apply Interest  
6: Switch to a Different Account  
0: To Exit

6

Which account would you like to switch to?

5

Account does not exist, please try again

Currently working with account: 0 Balance of \$4700.0

Please input a command:

1: Create a new Chequing Account  
2: Create a new Savings Account  
3: Deposit Funds  
4: Withdraw Funds  
5: Apply Interest  
6: Switch to a Different Account  
0: To Exit

0

**Your electronic submission (submitted via Desire2Learn) will consist of two files. Name your files YourName-fileName.extension, e.g. JohnSmith-as5.zip, JohnSmith-as5.pdf:**

1. A single pdf file containing a listing of the source code for the classes you have written. Also include your **driver**.
2. A zip file containing all your Java classes.