

The goal of this assignment is to get some practice using arrays. This assignment should be placed in package "hw02". In this assignment, we will extend the idea of the bank account class that we started in hw01. One of the changes we will make is to keep track of transactions performed on the account every month. To do that, we need a Transaction class.

Create a Transaction class with fields:

- starting balance – a field to keep track of the dollars and cents at the beginning of this transaction
- transaction type – a single character with a value of 'D' for deposit, 'W' for withdrawal, or 'I' for interest
- transaction amount – the dollars and cents amount of this transaction
- ending balance – a field to keep track of the dollars and cents at the end of this transaction.

The transaction class also needs a static array of month name strings, which I called "monthName". This array should have 12 entries, each with a three-character month name. For instance, monthName[0] should be "Jan".

The Transaction class should have a creator that takes three arguments... starting balance, type, and amount. (It should be able to calculate the ending balance based on the starting balance, the amount, and the type.)

We are going to print transaction reports from the Account class. To make things consistent and easy to grade, I will give you the three methods in the Transaction class we will use to print transaction reports, as follows (you may have to change the field names to match what you have chosen):

```
public void printHeader(int acct,int month, int year) {
    System.out.println(
" Transactions for account: " + acct + " for " + monthName[month] + " " + year);
    System.out.println(" | Start Bal. | T | Amount | End Bal. |");
    System.out.println(" | ----- + - + ----- + ----- |");
}
public void printLine() {
    System.out.println(String.format(" | %10.02f | %c | %10.02f | %10.02f |",
startingBalance,transactionType,transactionAmount,endingBalance));
}
public void printTrailer() {
    System.out.println(" | ----- + - + ----- + ----- |");
    System.out.println("");
}
}
```

Once you have coded the Transaction class, copy the Account class from hw01¹, and change the package to hw02. Add a new field for the log of transactions. The new field should be a reference to an array of Transaction objects. We will need to keep track of how many transactions we currently have, so add an integer field that will contain the index of the next unused log index. Also add integer fields to represent the number of the current month and the current year. The creator will instantiate the log field to an array of 100 (we are assuming less than 100 transactions per month) references to a Transaction object.

¹ After Monday, I will post my solution to hw01, and you can use that as a base if you had problems with hw01.

The creator should set the current month to 8 (which represents September when you start counting from zero), and the current year to 2012. Your creator should also create a new transaction to represent the initial deposit into the account. (The initial balance will be 0.0.) and put that in the log array.

Modify your deposit and withdraw methods in the Account class to create and log a new transaction to keep track of the deposit or withdrawal.

Finally, create a new printStatement method in the Account class. We are assuming the printStatement method will get invoked once every month. The printStatement method will do several things. First, it will calculate the interest to be added to the account for this month. Interest is 5% per year, therefore 5%/12 per month, and it is calculated monthly on the balance when the statement is printed. The interest should be added to the balance, and an interest transaction should be logged. Next, invoke the Transaction printHeader method. Then invoke the Transaction printLine method for every transaction currently used in the transaction log for this month. Then invoke the Transaction printTrailer method. Finally, printStatement should do some end of month cleanup... reset the transaction log index so that the next transaction gets created at index 0 of the log, and add 1 to the month (and if the month is greater than 11, set it back to zero and add one to the year.)

You should also copy and update the Tester class from hw01. Modify the Tester class to invoke the printStatement method on your accounts a few times.

Submission

Once all your Java code is written and tested, you will need to collect it all into a single file. Please create a zip file with the name "hw02_<userid>.zip", where <userid> is your bmail userid. This zip file should contain at least a hw02 directory that contains all your .java files for this assignment.

When you have your Java code collected in a single file, submit that file on MyCourses under "Content/Homework Submissions/HW02".

Grading

This assignment is worth 10 points.

- If there is a problem with your submission, such as the wrong file name or wrong archive format, but the code is in the submission, 2 points will be deducted.
- If your code does not compile, (i.e. if you get compiler errors) you will get eight points deducted.
- If your code produces any compiler warnings, 2 points will be deducted.
- If your code compiles, but produces incorrect results for *your* Tester program, 5 points will be deducted.
- The professor will write a Tester2 class with a main method that invokes your Account class methods. If you produce incorrect results using the professor's Tester2 class, 3 points will be deducted.
- If you submit after September 15 at 11:59 PM, 2 points will be deducted. No credit will be given if you submit after September 18 at 11:59 PM.
- Your code will be compared against all other submissions. If your code is similar to another student's (i.e. strong evidence that you both started from the same source), then both students will get a zero for this assignment.