CS 1400 - Assignment #7

Maximum Points: 20 pts

Topics:

- Inheritance
- Polymorphism

Coding Guideline (You will be graded on this)

1) Give identifiers semantic meaning and make them easy to read (examples numStudents, grossPay, etc).
2) Keep identifiers to a reasonably short length.
3) Use uppercase for constants. Use upper camel case for classes. Use lower camel case for all other identifiers (variables, methods, objects).
4) Use tabs or spaces to indent code within blocks (code surrounded by braces). This includes classes, methods, and code associated with ifs, switches and loops. Be consistent with the number of spaces or tabs that you use to indent.
5) Use white space to make your program more readable.
6) Use comments to explain how the parts of your program work.

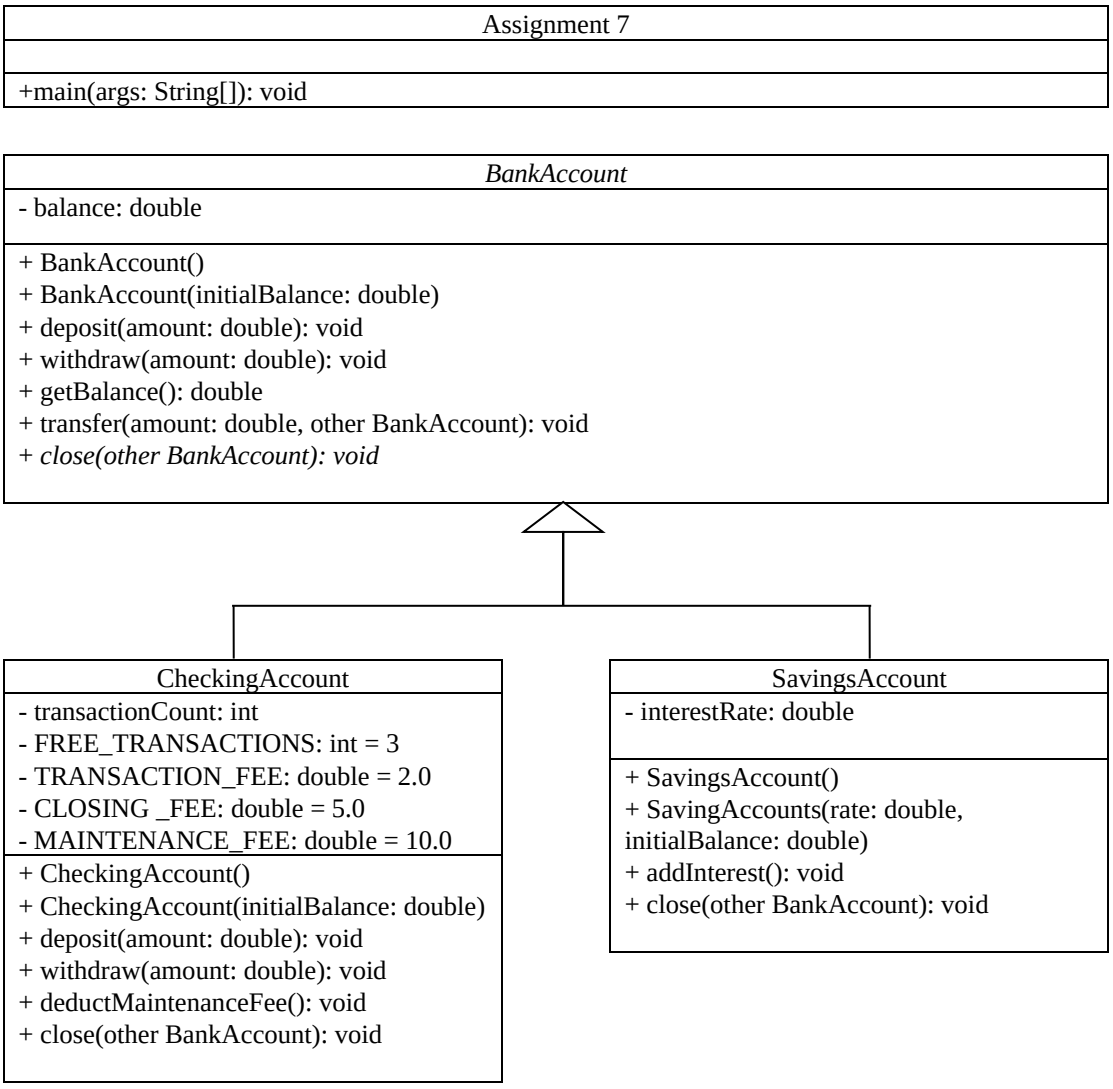
Important Note: All submitted assignments must begin with a descriptive block comment (multi-line comments) similar to the one shown below. It must contain your name and the other information illustrated. To avoid losing trivial points, make sure this comment header is included in every assignment you submit, and that it is updated accordingly from assignment to assignment.

```
/*-------------------------------------------------------------------------------
// AUTHOR: YOUR NAME
// FILENAME: TITLE OF THIS SOURCE FILE
// SPECIFICATION: DESCRIPTION OF THIS PROGRAM
// FOR: CS 1400 - ASSIGNMENT #7
// TIME SPENT: HOW LONG IT TOOK YOU TO FINISH THIS ASSIGNMENT
//*/
```

Part #1: Written Exercises: None

Part 2: Programming (20 points):

In this assignment, you will create three classes to simulate bank transactions: BankAccount, CheckingAccount, and SavingsAccount (there is no main method in those three classes). The provided driver program Assignment7.java should work correctly after you implement those three classes following the UML class diagram shown below. Make sure to implement all methods defined according to the specifications provided as well.



Class/Methods Specifications

Class: BankAccount	
Method	**Description of the Method**
public BankAccount ()	Constructs an empty bank account with a zero balance.
public BankAccount (double initialBalance)	Constructs a bank account with a given balance.
public void deposit (double amount)	Deposits money into the bank account.
public void withdraw (double amount)	Withdraws money from the bank account.
public double getBalance()	Gets the current balance of the bank account.
public void transfer(double amount, BankAccount other)	Transfers money from the bank account to another account.
public abstract void close (BankAccount other)	NA

Class: CheckingAccount	
Method	**Description of the Method**
public CheckingAccount ()	Constructs a checking account with no balance. The transaction count must be initialized.
public CheckingAccount (double initialBalance)	Constructs a checking account with a given balance. The transaction count must be initialized.
public void deposit (double amount)	Deposits money into the checking account. The transaction count must be incremented by one. If the transaction count is greater than the number of free transactions, it charges (withdraws) the transaction fee.
public void withdraw (double amount)	Withdraws money from the checking account. The transaction count must be incremented by one. If the transaction count is greater than the number of free transactions, it charges (withdraws) the transaction fee.
public void deductMaintenanceFee()	Charges (withdraws) the maintenance fee. The transaction count must be reset for a new month.
public void close (BankAccount other)	Closes the checking account and transfer the balance to another account. Charges (withdraws) the closing fee. Hint: the transaction count must be reset before the other operations.

Class: SavingsAccount	
Method	**Description of the Method**
public SavingsAccount ()	Constructs a savings account with no balance.
public SavingsAccount (double initialBalance)	Constructs a saving account with a given balance.
public addInterest ()	Adds the earned interest to the account balance.
public void close (BankAccount other)	Closes the savings account and transfer the balance to another account.

Use the provided program Assignment7.java which has the main method to test your classes. **You do NOT need to modify Assignment7.java.** The program will ask a user to enter the values that will be used in the bank transactions.

Sample run: user input is in RED **Bank Account Balance**

```
Inform the value to be withdrawn from mom's account.
50
```
```
Daddy|Mom|Nancy
 500   50    50
```

```
Inform the value to be withdrawn from mom's account.
40
```
```
Daddy|Mom|Nancy
 500   10    50
```

```
Inform the value to be transferred from daddy's to mom's account.
100
```
```
Daddy|Mom|Nancy
 400  110    50
```

```
Inform the value to be withdrawn from Nancy's account.
40
```
```
Daddy|Mom|Nancy
 400  110    10
```

```
Inform the value to be withdrawn from Nancy's account.
10
```
```
Daddy|Mom|Nancy
 400  110     0
```

Inform the value to be transferred from daddy's to Nancy's account.
50

Daddy|Mom|Nancy
350 110 50

Inform the value to be withdrawn from mom's account.
80

Daddy|Mom|Nancy
350 28 50

Inform the value to be withdrawn from Nancy's account.
30

Daddy|Mom|Nancy
350 28 18

Inform the value to be transferred from daddy's to mom's account.
100

Daddy|Mom|Nancy
250 126 18

Inform the value to be transferred from daddy's to Nancy's account.
50

Daddy|Mom|Nancy
200 126 66

Inform the value to be deposited in daddy's account.
500

Daddy|Mom|Nancy
500 324 66

Inform the value to be transferred from daddy's to Nancy's account.
100

Daddy|Mom|Nancy
400 324 164

Inform the value to be withdrawn from Nancy's account.
20

Daddy|Mom|Nancy
400 324 142

Inform the value to be withdrawn from Nancy's account.
30

Daddy|Mom|Nancy
400 324 110

Inform the value to be deposited in daddy's account
500

Daddy|Mom|Nancy
900 324 110

Inform the value to be withdrawn from mom's account.
150

Daddy|Mom|Nancy
900 172 110

Inform the value to be withdrawn from Nancy's account.
50

Daddy|Mom|Nancy
900 172 58

Inform the value to be transferred from mom's to Nancy's account.
50

Daddy|Mom|Nancy
900 120 106

Inform the value to be withdrawn from Nancy's account.
100

Daddy|Mom|Nancy
900 120 4

Inform the value to be deposited in daddy's account.
500

Daddy|Mom|Nancy
500 1013 4

Inform the value to be transferred from daddy's to Nancy's account.
50

Daddy|Mom|Nancy
450 1013 52

Inform the value to be withdrawn from Nancy's account.
15

Daddy|Mom|Nancy
450 1013 25

Inform the value to be withdrawn from Nancy's account.
15

Daddy|Mom|Nancy
450 1013 18

```
Inform the value to be withdrawn from mom's account.
100
Daddy|Mom|Nancy
 450  911  18

Inform the value to be withdrawn from mom's account.
200
Daddy|Mom|Nancy
 450  709  18

Inform the value to be transferred from mom's to Nancy's account.
200
Daddy|Mom|Nancy
 450  507  216

Mom's checking balance = $497.0
Nancy's checking balance = $206.0
Daddy's savings balance = $452.25
```

Helpful hints for doing this assignment:

- work on it in steps – write one method, test it with a test driver and make sure it works before going on to the next method
- always make sure your code compiles before you add another method
- your methods should be able to be called in any order

Submit your homework by following the instructions below:

****

- Upload your **Assignment7.java, BankAccount, CheckingAccount, and SavingsAccount** files on Gradescope.
- Assignment7.java should have the following in order:
  - In comments, the assignment Header described in "Important Note".
  - The working Java code requested in Part #2.
  - The Assignment7.java file must compile and run as you submit it. You can confirm this by viewing your submission results.
- Please check your code after you submit your assignment and make sure you are passing the two test cases.

Note: You may resubmit as many times as you like until the deadline. Only your last submission will be considered.

NO LATE ASSIGNMENTS WILL BE ACCEPTED. ALWAYS SUBMIT WHATEVER YOU HAVE COMPLETED FOR PARTIAL CREDIT BEFORE THE DEADLINE!