Student Name: Christopher Felix

Project: The Account Class

Problem Description: (The Account class) Design a class named Account that contains:
• A private int data field named id for the account (default 0).
• A private double data field named balance for the account (default 0).
• A private double data field named annualInterestRate that stores the current interest rate (default 0). Assume all accounts have the same interest rate.
• A private Date data field named dateCreated that stores the date when the account was created.
• A no-arg constructor that creates a default account.
• A constructor that creates an account with the specified id and initial balance.
• The accessor and mutator methods for id, balance, and annualInterestRate.
• The accessor method for dateCreated.
• A method named getMonthlyInterestRate() that returns the monthly interest rate.
• A method named withdraw that withdraws a specified amount from the account.
• A method named deposit that deposits a specified amount to the account. Draw the UML diagram for the class. Implement the class. Write a test program that creates an Account object with an account ID of 1122, a balance of $20,000, and an annual interest rate of 4.5%. Use the withdraw method to withdraw $2,500, use the deposit method to deposit $3,000, and print the balance, the monthly interest, and the date when this account was created.

Analysis: (Describe the problem including input and output in your own words.)

The program should be a class called “Account” that simulates a bank account. It includes private fields for account details such as “id” and “balance”. The program should feature a no-argument constructor and a constructor that initializes the account with a given id and balance. Moreover, there are methods to handle depositing and withdrawing money, calculating monthly interest, and retrieving the account details.

Input: account ID, balance, interest rate, and transaction amounts

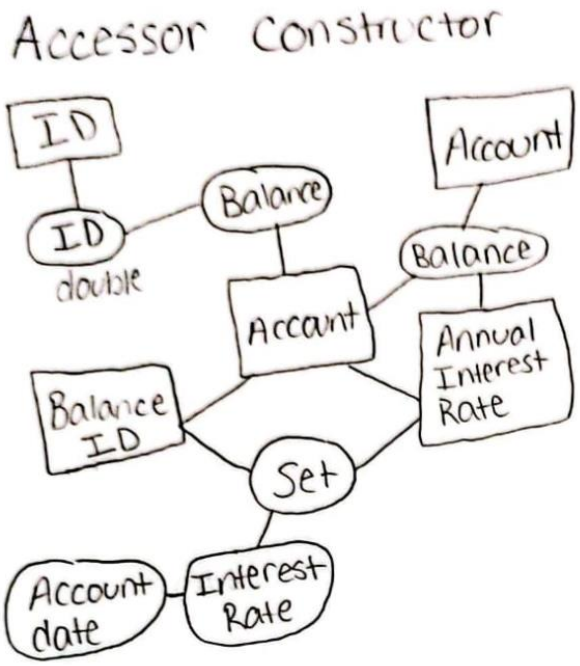
Outputs: account’s updated balance, monthly interest, and the account’s creation date

Design: (Draw an UML class diagram for the Account class.)

```
/**********************************************************
*              Account                  *
*---------------------------------------------------------*
* -id: int                              *
* -balance: double                      *
* -annualInterestRate: double           *
* --------------------------            *
* -dateCreated: String                  *
* +Account()                            *
* +Account(newid: int, newBalance: double)          *
* +setId(newId: int)                    *
* +setBalance(newBalance: double)               *
* +setAnnualInterestRate(newAnnualInterestRate: double) *
* +getId(): int                         *
* +getBalance(): double                 *
* +getAnnualInterestRate(): double              *
* +getDateCreated(): String             *
* +getMonthlyInterestRate(): double             *
* +getMonthlyInterest(): double             *
* +withdraw(amount: double)             *
* +deposit(amount: double)              *
**********************************************************/
```

The “-” symbol indicates private visibility, meaning these members can only be accessed within the class itself.

The “+” symbol indicates public visibility, meaning these members can be accessed from outside the class.



Testing: (Describe how you test this program)

Initial test case: Create an Account object with an account ID of 1122 and a balance of $20,000. Next, set the annual interest rate to 4.5% and verify that the account information is correct.

Withdraw test: Ensure that the balance has decreased by the correct amount ($20,000 - $2,500 = $17,500).

Deposit test: Ensure that the balance has increased by the correct amount ($17,500 + $3,000 = $20,500).

Interest calculation: Calculate the monthly interest based on the 4.5% annual rate and confirm that the calculation matches expectations using the formula.

Account creation date: Ensure that the “dateCreated” field accurately reflects the date when the account was instantiated.

Submit the following items:
1. Print this Word file and Submit to me before the class on the due day
2. Compile, Run, and Submit