**Introduction to Java Assignment 9**

**Due Date: Please Watch D2L deadline for submission date**

**Bank Account Class**

In this program we simulate small part of an account in a bank. The class has the following private instance data members, which are all declared private. Their description, data types, and suggested names for them are also given.

|  |  |  |
| --- | --- | --- |
| Instance data member Description | Instance data member data type | Suggested variable name |
| Account number associated with the bank account | integer | ActNum |
| Last Name of the account holder | String | LastName |
| First Name of the account holder | String | FirstName |
| Current Balance in dollars in the account | double | balance |
| **Table 1** | | |

Proceed in the following manner to code either using textpad or Eclipse.

1. Generate a Java class called BankAccount.
2. Add the data members (all private) described in the table 1 to the class.
3. Write an explicit constructor, in above class, which takes four arguments, each one of which maps to the four class instance data members. The suggested header (first line) for the constructor is shown on next page:

public BankAccount(int act, String first, String last, double bal)

{

Class member ActNum will be equal to act

Class member FirstName will be equal to first.

Class member LastName will be equal to last

Class member balance will be equal to bal.

//body of explicit constructor

}

1. Write a print method in class BankAccount whose suggested header is given below.

public void print()

{

//body of print method

}

This method prints all class data member values to console. Use printf for currency formatting and output to console.

1. Write another class (suggested name: MainClass), and write a main method in it.
2. In main method call the explicit constructor you wrote in step 3 and pass some hard coded values as arguments to this constructor. This way you are creating an object of type BankAccount in your main method. Suggested values for example are as follows:

Account number: 5623

First Name: Jim

Last Name: Jones

Balance: 100.89

1. Call the print method for the BankAccount object you created in step 6 which when you run the program would print same values that you passed to explicit constructor as arguments. This will confirm that your explicit constructor is working correctly. If not then fix all errors at this point.
2. Back to the BankAccount class: Now write a method called deposit which simulates depositing money in a bank account. Deposit method will take a double type argument which is the amount of money customer wishes to deposit to their current account. Thus after deposit method is executed the customer’s balance in the account will increase by the amount of money deposited. The suggested header for the method deposit is on next page.

public void deposit(double money)

{

//body of deposit method

}

1. Go back to main method in MainClass and test to make sure that your deposit method works correctly. If not then fix its errors before continuing.
2. Finally write a method called withdraw in BankAccount class which simulates the withdrawal of money from a bank account. As you know full well that at best you can only withdraw only that much money from your account that equals your balance. This rule must be enforced by the method withdraw. It should allow withdrawal of money, and readjust the balance downwards, if current balance is equal to or exceeds the withdrawal amount. Otherwise it refuses the transactions by saying “Insufficient funds” and would not change the balance. In that sense it acts as a mutator that mutates the balance data member only if amount of withdrawal is less than or equal to balance. The suggested header for the method withdrawal is below.

public void withdrawal(double money)

{

//body of withdrawal method

}

1. Go back to your main method in MainClass and now test to make sure that withdrawal method is working correctly. This requires that you do two tests one with amount of withdrawal less than the balance and other more than the balance. The transaction must be refused when withdrawal amount exceeds the balance.

**Client Main Method**

By now you have done enough tests to ascertain that your class (presumably) has no bugs. Now we do a program which demonstrates as how client can use our BankAccount class. You can create a third class in your project. (Suggested name: ClientMainClass). In this class you do a menu driven program that will have following menu items.

1. Create an instance of Bank Account from keyboard data entry.
2. Print the details of Bank Account instance to console.
3. Print details of Bank Account instance to an output file.
4. Make a deposit.
5. Make a withdrawal.
6. Print current balance.
7. Print account holder’s full name.
8. Exit

I will be happy to show you a working program so you can grasp all specifications and program requirements.