**CSEC 323: Homework #4 – Designing a Secure Bank Account Class**

For Homework #4, you will develop some of the elements required for Project 1. Given the problem definition for Project 1…

1. **Document the public interface and data model for the Bank Account and Transaction classes.**

| **Public Interface and Data Model for the Bank Account Class** | | | |
| --- | --- | --- | --- |
| **Task** | **Method** | **Data Needed** | **Return Value** |
| Initialize a new bank account - ZN | \_\_init\_\_(str firstIn, str lastIn, float balanceIn = 0.0) | First name (String)  Last name (String)  Next available account number (int)  Initial balance (float) | Bank Account object |
| Accessor/getter method for the overdraft fee - AP | getOverdraft() | Overdraft fee (float) | Overdraft fee (float) |
| Accessor/getter method for the interest rate - AP | getIntRate() | Interest rate (float) | Interest rate (float) |
| Accessor/getter method for the next available account number - ZN | getNextAccount() | Next available account number (int) | Next available account number (int) |
| Accessor/getter method for the first name - AP | getFirst() | First name (String) | First name (String) |
| Accessor/getter method for the last name - AP | getLast() | Last name (String) | Last name (String) |
| Accessor/getter method for the account number - ZN | getAccountNumber() | Assigned account number (int) | Assigned account number (int) |
| Accessor/getter method for counter for when the account is overdrawn - AP | getOverdrawnCount() | Overdrawn counter value (int) | Overdrawn counter value (int) |
| Accessor/getter method for the account balance - AP | getBalance() | Current account balance (float) | Current account balance (float) |
| Mutator/setter method for the overdraft fee - AP | \_setOverdraft(float newFee) | Overdraft fee (float) | N/A |
| Mutator/setter method for the interest rate - AP | \_setIntRate(float newRate) | Interest rate (float) | N/A |
| Mutator/setter for the first name - AP | setFirst(str first) | First name (String) | N/A |
| Mutator/setter for the last name - AP | setLast(str last) | Last name (String) | N/A |
| Display the account details - AP | \_\_repr\_\_() | First name (String)  Last name (String)  Assigned account number (int)  Initial balance (float)  Results from call to transactionList(), which requires a list of transactions for the account (List or Set) | String (Contains all account information with string formatting) |
| Print the list of transactions for an account (the accessor/getter method for the transaction list) - AP | transactionList() | List of transactions for the account (List or Set) | String (Contains all transactions with string formatting) |
| Calculates and adds interest to the account (if applicable) - HP | calc\_interest() | Current balance (float)  Interest rate (float)  List of transactions for the account (List or Set) | Boolean (True if interest needs to be added, False if not) |
| Transfer an amount to one account from another account -BS | transfer(float amount, BankAccount otherAccount) | Amount being transferred (float)  Account the money is being transferred to (BankAccount object)  List of transactions for the account transferring (List)  Results from call to deposit(), which requires the amount being deposited (float), current account balance of the account being deposited to (float), and the list of transactions for the account being deposited to (List or Set)  Results from call to withdraw(), which requires the amount being withdrawn (float), the current balance of the account being withdrawn from (float), the overdraft fee (float), the overdrawn count of the account being withdrawn from (int), and the list of transactions for the account being withdrawn from (List) | Boolean (True if transfer is possible, False if not) |
| Deposit an amount into the account - BK | deposit(float amount) | Amount being deposited (float)  Current account balance (float)  List of transactions for the account (List or Set) | Boolean (True if the amount passed in was valid and thus the deposit is valid, False if not) |
| Withdraw an amount from the account (if possible) - BK | withdraw(float amount) | Amount being withdrawn (float)  Current balance (float)  Overdraft fee (float)  Overdrawn count (int)  List of transactions for the account (List or Set) | Boolean (True if withdrawal is possible, False if not) |
| Determines if two Bank Account objects are equal - AP | \_\_eq\_\_(self, otherAccount) | Second account that is being compared (BankAccount object)  Assigned account number of the first BankAccount object being compared (int)  Assigned account number of the second BankAccount object being compared (int) | Boolean (True if the BankAccount objects are equal, False if not) |

| **Public Interface and Data Model for the Transaction Class** | | | |
| --- | --- | --- | --- |
| **Task** | **Method** | **Data Needed** | **Return Value** |
| Initialize a new transaction | \_\_init\_\_(str type, float amount) | Results from \_setDate(), which requires access to the current day (int), current month (int), and current year (int)  Transaction amount (float)  Transaction type (String)  Next available transaction number (int) | N/A |
| Determines if two Transactions are equal | \_\_eq\_\_(Transaction other) | Second transaction that is being compared (Transaction object)  Transaction amount of the first transaction object (float)  Transaction amount of the second transaction object (float)  Date of the first transaction object (String)  Date of the second transaction object (String)  Transaction type of the first transaction object (String)  Transaction type of the second transaction object (String) | Boolean (True if the Transaction objects are equal, False if not) |
| Determines if two Transactions are not equal | \_\_ne\_\_(Transaction other) | Second transaction that is being compared (Transaction object)  Transaction amount of the first transaction object (float)  Transaction amount of the second transaction object (float)  Date of the first transaction object (String)  Date of the second transaction object (String)  Transaction type of the first transaction object (String)  Transaction type of the second transaction object (String) | Boolean (True if the Transaction objects are not equal, False if they are equal) |
| Adds two transaction amounts together | \_\_add\_\_(Transaction other) | Second transaction that is being added (Transaction object)  Transaction amount of the first transaction object (float)  Transaction amount of the second transaction object (float) | Float |
| Subtracts one transaction amount from another | \_\_sub\_\_(Transaction other) | Second transaction that is being subtracted (Transaction object)  Transaction amount of the first transaction object (float)  Transaction amount of the second transaction object (float) | Float |
| Implements the sum() function that will sum a list of the Transactions | \_\_radd\_\_(Transaction other) | Amount that has been summed so far (float)  Transaction amount of the transaction object (float) | Float |
| Accessor/getter method for the transaction day | getDay() | Current day (int) | Int |
| Accessor/getter method for the transaction month | getMonth() | Current month (int) | Int |
| Accessor/getter method for the transaction year | getYear() | Current year (int) | Int |
| Accessor/getter method for the transaction amount | getAmount() | Transaction amount (float) | Float |
| Accessor/getter method for the transaction date | getDate() | Current date (contains the current day (int), current month (int), and current year (int)) | String |
| Accessor/getter method for a specific transaction number | getTNumber() | Transaction number (int) | Int |
| Accessor/getter method for the transaction type | getTType() | Transaction type (String) | String |
| Accessor/getter method for the next available transaction number | getNextTNumber() | Next available transaction number (int) | Int |
| Display the details of a transaction | printTransaction() | Date of the transaction (String)  Transaction amount (float)  Transaction type (String)  Transaction number (int) | String (Contains transaction information with string formatting) |
| Prints all of the transaction instance variables in a human-readable form | \_\_str\_\_() | Transaction number (int)  Transaction amount (float)  Date of the transaction (String)  Transaction type (String) | String (Contains the formatted, human-readable transaction information) |
| Prints all of the transaction instance variables in a machine-readable form | \_\_repr\_\_() | Transaction number (int)  Transaction amount (float)  Date of the transaction (String)  Transaction type (String) | String (Contains the formatted, machine-readable transaction information) |
| Private helper method that sets the date for a transaction | \_setDate() | Date of the transaction (String)  Day of the transaction (int)  Month of the transaction (int)  Year of the transaction (int) | N/A |

1. **Document the responsibilities for the Bank Account and Transaction classes.**

The Bank Account class needs to know…

* The first name of the person who owns the Bank Account (String)
* The last name of the person who owns the Bank Account (String)
* The initial balance being held within the Bank Account (float)
* The current account number (int)
* The next available account number for a new Bank Account (int)
* The overdraft fee amount for all Bank Accounts (float)
* The interest rate for all Bank Accounts (float)

The Bank Account class needs to have implemented (do)…

* Accessor (getter) methods for all instance variables
* Mutator (setter) methods for all mutable instance variables
* Display the details of an account
* Deposit an amount into an account
* Withdraw an amount from an account (if possible)
* Calculate and add interest to the account (if applicable)
* Transfer an amount from one account to another account
* Print the list of transactions for an account
* Determine if two accounts are equal/not equal to one another

The Transaction class needs to know…

* The current day (int)
* The current month (int)
* The current year (int)
* The transaction amount (float)
* The transaction type (String)
* The current transaction number (int)
* The next available transaction number (int)

The Transaction class needs to have implemented (do)…

* Accessor (getter) methods for all instance variables
* Mutator (setter) methods for all mutable instance variables
* Get the transaction date (current date) from the system
* Display the details of a transaction
* Determines if two Transactions are equal/not equal
* Add/subtract transactions together
* Sum all transactions within a transaction list
* Set the date for a transaction

1. **Document the definitions for the Bank Account and Transaction classes.**

The class definitions for the Bank Account class are as follows:

* Data
  + First name: String that represents a person’s first name; should be valid (potentially check for not empty string; 1 – 25 alphabetical characters with no special characters)
  + Last name: String that represents a person’s last name; should be valid (potentially check for not empty string; 1 – 40 alphabetical characters with no special characters)
  + Next available account number: Int that represents the next available account number to be given to a new Bank Account
  + Current account number: Int that represents the account number for a Bank Account
  + Initial balance: Float that represents the initial deposit/withdrawal value of a new Bank Account
  + Current balance: Float that represents the overall value within a Bank Account
  + Overdraft fee: Float that represents the overdraft fee for all Bank Accounts
  + Interest rate: Float that represents the interest rate for all Bank Accounts
  + Overdrawn count: Int that represents how many times a Bank Account is overdrawn; starts at 0
  + Transaction list: List or set that holds all transactions for a Bank Account; always starts with an initial deposit/withdrawal depending on how the Bank Account is initialized
* Queries
  + getOverdraft(): Returns the overdraft fee amount as a float
  + getIntRate(): Returns the interest rate as a float
  + getNextAccount(): Returns the next available account number as an int
  + getFirst(): Returns the first name of the account holder as a String
  + getLast(): Returns the last name of the account holder as a String
  + getAccountNumber(): Returns the account number of the Bank Account as an int
  + getOverdrawnCount(): Returns the amount of times the Bank Account has been overdrawn as an int
  + getBalance(): Returns the total held balance of the Bank Account as a float
* Commands
  + \_setOverdraft(float newFee): Private method that sets a new overdraft fee as a float for all Bank Accounts
  + \_setIntRate(float newRate): Private method that sets a new interest rate as a float for all Bank Accounts
  + setFirst(str first): Sets a new first name as a String
  + setLast(str last): Sets a new last name as a String
  + \_\_repr\_\_(): Displays the account details with String formatting
  + deposit(float amount): Deposits an amount into the account
  + withdraw(float amount): Withdraws an amount from the account (if possible)
  + calcInterest(): Calculates and adds interest to the account (if applicable)
  + transfer(float amount, BankAccount otherAccount): Transfers an amount to one account from another account (if applicable)
  + transactionList(): Prints the list of transactions for an account (the accessor/getter method for the transaction list)
  + \_\_eq\_\_(BankAccount otherAccount): Determines if two BankAccount objects are equal to one another

The class definitions for the Transaction class are as follows:

* Data
  + Day: Int that represents the day of the month; must be between 1 – 31
  + Month: Int that represents the month of the year; must be between 1 – 12
  + Year: Int that represents the year; must be greater than or equal to 2022
  + Amount: Float that represents the amount of the transaction
  + transactionNumber: Int that represents the transaction number nextTransactionNumber: Int that represents the next available transaction number; the initial value starts at 100
* Queries
  + getDay(): Returns the day of a transaction as an int
  + getMonth(): Returns the month of a transaction as an int
  + getYear(): Returns the year of a transaction as an int
  + getAmount(): Returns the amount of the transaction as a float
  + getDate(): Returns the day, month, and year of a transaction with String formatting
  + getTNumber(): Returns the transaction number of the transaction as an int
  + getTType(): Returns the transaction type of the transaction as a String
  + getNextTNumber(): Returns the next transaction number to be assigned to a new transaction as an int
* Commands
  + \_\_eq\_\_(Transaction other): Determines if two Transactions are equal
  + \_\_ne\_\_(Transaction other): Determines if two Transactions are not equal
  + \_\_add\_\_(Transaction other): Adds two transaction amounts together
  + \_\_sub\_\_(Transaction other): Subtracts one transaction amount from another
  + \_\_radd\_\_(Transaction other): Implements the sum() function that will sum a list of the Transactions
  + printTransaction(): Prints the day, month, year, amount, transaction number, and transaction type with String formatting
  + \_\_str\_\_(): Prints all of the transaction instance variables in a human-readable form
  + \_\_repr\_\_(): Prints all of the transaction instance variables in a machine-readable form
  + \_setDate(): Private helper method that sets the date for a transaction