

Assignment

Design by Contract, Preconditions, Postconditions, Class Invariants, Loop Invariants

1. Description of the Assignment

1.1 Read if you required to

Read the book if you need to clear the concept. Book is in the directory named as Book.pdf

1.2. Checking Accounts and Credit Accounts Banking System

You are required to apply the concepts of Design by Contract, preconditions, postconditions, class invariants, and loop invariants that were presented in the book.pdf Chapter 11, in the design, implementation, testing, and documentation of a Checking Accounts and Credit Accounts Banking System with the following requirements:

- A. The Checking Accounts and Credit Accounts Banking System includes at least 3 different kinds of accounts: checking accounts, credit accounts, and demand loan accounts.

A1. Checking accounts

For each checking account one of the following 3 options regarding overdraft protection must be selected:

Option 1. No Overdraft Protection: with this option, if a withdrawal from the checking account would cause the balance to be less than 0, then the withdrawal will be declined, and a Non-Sufficient Funds (NSF) penalty will be charged.

Option 2. Pay Per Use Overdraft Protection

Option 3. Monthly Fixed Fee Overdraft Protection.

The rules for checking accounts with these 3 overdraft protection options are explained in detail in the document rules

CIBC_Overdraft_Protection_Service_Addendum.pdf

A2. Credit accounts

For credit accounts the following rules related to the credit limit of a credit account will be enforced:

Rule 1. High Credit Limit with Credit Limit Exceed Penalty: according to this rule, if the credit limit is greater than \$1000, and a withdrawal from the credit account would cause the sum of the balance and credit limit to be less than 0, then the withdrawal will be declined, and a Credit Exceed Limit (CLE) penalty will be charged.

Rule 2. Low Credit Limit with No Credit Limit Exceed Penalty: according to this rule, if the credit limit is less than or equal to \$1000, and a withdrawal from the credit account would cause the sum of the balance and credit limit to be less than 0, then the withdrawal will be declined, but no Credit Exceed Limit (CLE) penalty will be charged.

A3. Demand loan accounts

A demand loan account corresponds to a demand loan as described in the document CIBC_Overdraft_Protection_Service_Addendum.pdf

B. The following operations are required for checking and credit accounts:

B1. withdrawAmount()

Withdraw a specified amount from a specified account

B2. depositAmount()

Deposit a specified amount into a specified account.

B3. createAccount()

Create an account of a specified type.

B4. cancelAccount()

Cancel a specified account.

B5. suspendAccount()

Suspend a specified account.

B6. reactivateAccount()

Reactivate a specified account.

B7. getbalance()

Return the current balance of a specified account.

B8. terminateAccount()

Terminate a specified account and convert the Indebtedness to a demand loan.

B9. setOverdraftOption():

Set the overdraft protection option of a specified checking account to one of the three overdraft protection options.

B10. setLimit()

Set the Overdraft Limit Amount of a specified checking account to a specified amount; or set the Credit Limit of a specified credit account to a specified amount.

B11. transferAmount()

Transfer a specified amount from one specified account to another specified account.

- C. In addition, each of the operations B1-B11 must also record relevant information about the operation in a record of type AccountActivity, and add that record to a sequence of records called accountLog where each record is of type AccountActivity. Every time before shutting down the system, the sequence called accountLog must be saved to a file.

The following operations are required related to the accountLog where each record is of type AccountActivity:

C1. sortAccountLog():

- use the Insertion Sort algorithm to sort all the records in the accountLog by Customer Security Number (SIN), such that any record with a smaller SIN will be ordered before any record with a larger SIN, and for records with the same Customer Security Number any record with an earlier Date and Time will be ordered before any record with a later Date and Time

C2. processAccountLogEndOfDay():

- process records in the accountLog at the end of each business day according to the rules in the document

CIBC_Overdraft_Protection_Service_Addendum.pdf
and Rule 1 and Rule 2 for credit accounts above.

C3. processAccountLogEndOfMonth():

- process records in the accountLog at the end of each calendar month according to the rules in the document

CIBC_Overdraft_Protection_Service_Addendum.pdf
and Rule 1 and Rule 2 for credit accounts above.

C4. saveAccountLog():

- save the accountLog to a file.

C5. retrieveAccountLog()

- retrieve the previously saved accountLog from a file

- D. Here is a list of assumptions that you may make in addition to the rules in the document CIBC_Overdraft_Protection_Service_Addendum.pdf

(1) Each customer has a unique Social Security Number that is provided by the customer when an account is created for the customer.

(2) Whenever a customer makes a request to cancel an account, the account is always treated as actually cancelled at the end of the calendar month. Whenever the bank terminates an account, the account is always cancelled immediately.

(3) Each customer cannot possess more than one checking account and cannot possess more than one credit account at any point in time.

(4) Interest on negative balances for both checking accounts and credit accounts are calculated using the method described in the document CIBC_Overdraft_Protection_Service_Addendum.pdf

(5) The Credit Exceed Limit (CLE) penalty is \$29.

(6) A transfer operation which transfers a specified amount from a specified first account to a specified second account can be treated as withdrawing the specified amount from the specified first account, then depositing the specified amount into the specified second account.

1.3. Requirements regarding the design, implementation, testing, and documentation of the software in the Checking Accounts and Credit Accounts Banking System

- (a) You must apply the concepts regarding Design by Contract, preconditions, postconditions, class invariants, and loop invariants described in the Book.pdf in the design, implementation, testing, and documentation of the Checking Accounts and Credit Accounts Banking System to the greatest extent possible.

In addition, your Report must include a Summary Section, in which you must provide a summary of how you applied the concepts regarding Design by Contract, reconditions, postconditions, class invariants, and loop invariants in the design, implementation, testing, and documentation of the Checking Accounts and Credit Accounts Banking System.

- (b) When designing the software to implement the Checking Accounts and Credit Accounts Banking System, you must carefully choose appropriate data structures. Furthermore, in your report/documentation you must justify and explain why you chose each particular data structure.
- (c) In your program code and in your report/documentation, you must use appropriate notation to write the class invariant for each class, the precondition and postcondition for each routine/method in each class, and the loop invariant for each loop in each method/routine. *The class invariants, preconditions, postconditions, and loop invariants must be Boolean expressions which incorporate the actual variables and constants that are used in your program code.*
- (d) In your report/documentation, you must provide detailed explanation and justification for the class invariant for each class, the precondition and postcondition for each routine/method in each class, and the loop invariant for each loop in each method/routine.
- (e) The Checking Accounts and Credit Accounts Banking System should adhere as closely as possible to all the rules defined in the document CIBC_Overdraft_Protection_Service_Addendum.pdf.
- (f) Additional requirements
 - (f1) You must make sure that your code has very detailed comments.
 - (f2) You must make sure that your code compiles correctly.
 - (f3) You must make sure that your code does not generate non-recoverable exceptions.
 - (f4) You must make sure that your code is able to handle incorrect input.
 - (f5) You must describe in detail any problems or difficulties that you had encountered, and how you solved or were able to overcome those problems or difficulties in the report.

2. Platform on Which the Checking Accounts and Credit Accounts Banking System is to be Implemented

The programs should be implemented using the Java programming language

3. What to Submit

1. A written report that identifies and addresses all the important aspects and issues in the design, implementation, testing, and documentation of the software for the problem described above.
2. The Java source programs.
3. A “Test_output” file containing the output of any testing.
4. A “README” file explaining how to compile and run your group’s program.

4. Evaluation of the Assignment

4.1. The report part of your assignment (70%) will be evaluated according to:

- (a) How well you have satisfied the requirements specified in Section 1.3 above.
- (b) How well you have justified your design decisions.
- (c) The quality of your design.
- (d) How well you have designed and explained the testing.
- (e) The clarity, and readability of the report.

4.2. The program and testing part of your assignment (30%) will be evaluated according to:

- (a) The quality of the design and implementation of your programs.
- (b) The quality of the testing of your programs.
- (c) Whether your programs satisfy the Additional Requirements in section 1.3(f) above

5. Notes

Obviously, there are many other possible details of the Checking Accounts and Credit Accounts Banking System that have been left unspecified. It is your responsibility to make appropriate design, implementation, testing, and documentation choices concerning the unspecified details of the Checking Accounts and Credit Accounts Banking System, and justify those decisions in your report.