University of Vavuniya, Sri Lanka
Faculty of Technological Studies
Department of ICT
TICT1224(P) - Object Oriented Programming (Practical)
In-course Assessment Examination – 01

Time Allocated: 75 minutes 05th September 2023

You are requested to submit your answers in a zipped folder named with your registration number (Eg: 2020ICTS##) only with the .java files

1. Consider the following class diagram.

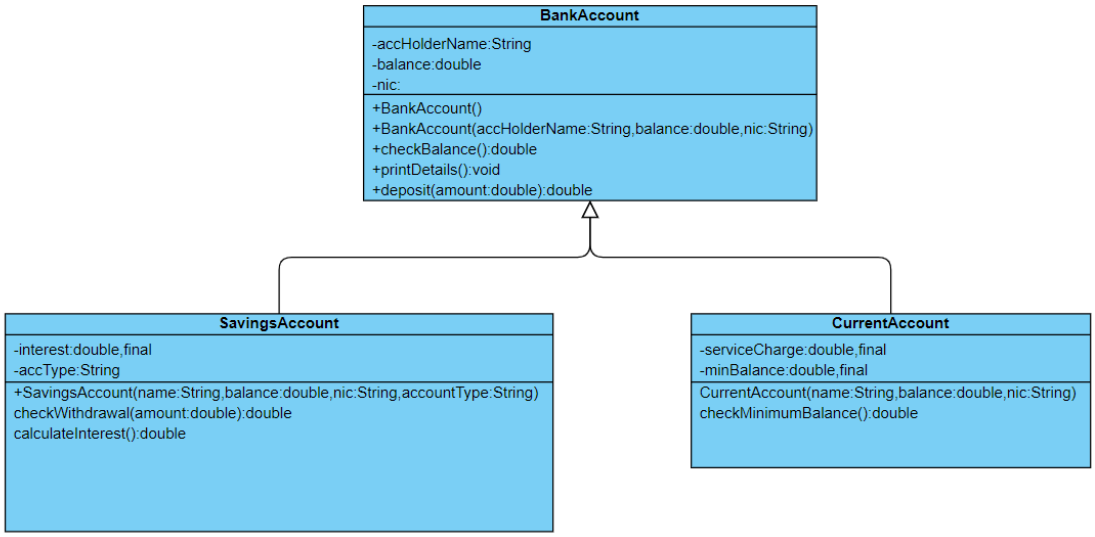

Figure 1:UML Class Diagram

You are required to map the above class diagram into a computer program using Java programming language. Please note that + indicates the public class members and - indicates the private class members.

A. Create a class named *BankAccount* which stores the *name, balance* and *nic* of a particular BankAccount.
B. Derive two classes named as *SavingsAccount* and *CurrentAccount* from the class *BankAccount*.
C. Store the data members of the derived classes according to the class diagram given above.
D. Create constructors in all three classes to initialize their respective data members as given in the class diagram above.
E. Call the super class constructor in the derived class constructors' using the appropriate keywords

The details of methods inside each class are given below.

Class: BankAccount

(a) The *checkBalance*() method should display the account balance.

(b) The *printDetails*() method in the super class should print the name of the *account holder*, *nic* and the *balance* details.

(c) *deposit*() method is used to accept deposits from the customer as the parameter of the method and update the account balance based on the following equation.

balance=balance+amount

Class: Savings Account

(a) Create a constant variable *interest* inside the class *SavingsAccount* and initialize the value 10.0 to it.

(b) Create a method *calculateInterest*() to update the balance according to the interest. The equation is given below.

balance=balance+(balance(interest/100))*

(c) Create another method *checkWithdrawal*() to permit withdrawal and to update the *balance*.

The system should prompt a message "Insufficient Balance!", if the *balance* is less than the withdrawal *amount* , and prompt "Successfully Withdrawed" instead.

The balance should be updated upon successful withdrawal based on the following equation.

balance=balance-amount

Class: Current Account

(a) Create *CurrentAccount* class where current account holders should maintain a minimum balance and if balance falls below this level a service charge is imposed (Service charge will be deducted from the current balance).

(b) To check for the minimum balance, create two constants as *servicecharge* and *minbalance* and initialize these constants as 150.00 and 1500.00 respectively.

(c) Using the *checkMinium*() method, check for the minimum balance, impose the service charge and update the balance as follows.

- If balance is less than *minbalance*, then the service charge should be deducted from the *balance* (balance=balance-servicecharge).

F. Create a separate class named *BankApp* which contains the main method.

G. Create two objects for both the following users, using the relevant classes and call the relevant methods based on the actions performed over their account.

Sample Data

1. Account type: Savings Account
   Account holder name: Kamal
   Balance: 10000.00

   The following actions are performed over his account.
   - Check for the balance
   - Deposit 5000.00
   - Withdraw 1000.00
   - Calculate the interest over the current balance.
   - Check for the balance

2. Account Type: Current Account
   Account holder name : Amal
   Balance: 1000.00

   The following actions are performed over his account.
   - Check for the balance
   - Deposit 500.00
   - Check for the minimum balance
   - Check the current balance

[100%]