

CPSC 233 – Individual Assignment 4 – Requirements

To be able to work on this assignment, you will need a functional version of the BankAccount and Customer class from last week's assignment. Your first task is to get this working.

Similar to previous assignments, you are expected to infer some of the requirements from the unit tests.

Add a third class called SavingsAccount which is a subclass of BankAccount. In addition to the methods and data provided by the parent class, SavingsAccount should include:

annualInterestRate – an instance variable of type double, which indicates an annual interest rate, which is between 0 and 1 (representing 0% to 100%).

minimumBalance – an instance variable of type double which represents the minimum balance required for the account. Withdrawals that would allow the balance in the account to go below this account should not be allowed. This will require you to **override the withdraw method** from the parent class. The child class (SavingsAccount) should only verify that there are sufficient funds in the account for the withdraw. If there is, it should invoke the withdraw method in the parent class.

depositMonthlyInterest() – a method which, on a monthly basis, calculates the amount of interest earned on the balance in the SavingsAccount, and adds this to the balance.

SavingsAccount should not contain any additional instance variables or methods. To update the balance in the account, the methods in SavingsAccount should call the methods in the parent class.

A JUnit test is provided below for SavingsAccount. (The classes BankAccount and Customer are unchanged so no additional tests are provided for this.) This tests not only the functionality but also the design of the class.

Submit only your SavingsAccount class to WebCAT. There are 11 tests. To qualify for the full team grade you must pass at least 9/11 tests.