

## CPSC 233 – Team Assignment 6 – Requirements

As a team, complete a solution for the individual assignment that you'll use for your team submission. Then update BankAccount by:

- Adding boolean method sufficientFunds that takes a double and checks if there is sufficient funds in the account to withdraw the specified amount.
- Update methods withdraw and transfer to use this boolean method before doing a withdrawal.

Create a class called ChequingAccount as a subclass of BankAccount such that it has

- Two instance variables: overdraftFee and overdraftAmount both of type double.
- Setter and getter methods for both instance variables. Both must be non-negative numbers.
- Create a constructor that takes an account holder (type Customer), start balance (type double) and overdraft fee (type double) as arguments. (No other constructors required.)
- Override the following methods from BankAccount:
  - sufficientFunds such that it returns true if the amount specified (for withdrawal) will result in a balance that might be less than zero, but never more than overdraftAmount below zero.
  - withdraw such that it updates the amount to withdraw to add the overdraftFee if the withdraw would result in a negative balance. (It should invoke withdraw in the parent to do the actual withdrawal.)
  - getMonthlyFeesAndInterest such that it returns 0 if the balance is greater than or equal to zero but returns 20% of the balance if the balance is less than zero.

### Rubric

	2 marks	1 mark	0 marks
Functionality	All tests in the three test classes pass.		One or more tests fail.
Design,	There is no code/instance variable duplication. Constructors and overridden methods use 'this' and super where appropriate. All instance variables are private. Methods access as required. Components are abstract as required.	Some code duplication or access is public when it should be private/protected.	Did not use abstract as required, some code duplication, instance variables from parent are repeated in child class.
Legibility	Code is easy to read.	Some improvement possible but overall code is legible.	Code is difficult to read.
Documentation	Class and methods are javadoc'd. Inline documentation provided where needed i	Documented but not to correct audience.	Javadoc missing or inline documentation missing when some code is complex.