CS 485 Assignment 2 01_Bank

Name	Grade ____ / 45
Builds without errors (-40% if not)	
Builds without warnings (- 10% if not)	
Hardcopy, stapled, color, double sided (-1 pt if not)	
Successful Execution (~40%) 18 pts	
Test 1: Posted Test Case (4 pts) all or nothing	
Test 2: 1 Savings Account (4 pts) S 1 10000 0.01 200 5000 # 1/2 pt per P P W 1 5001 # 49.99 P M # Fee incurred (49.99 -2.00)* 1.01 = 48.46 P D 1 1 # 48.46 + 0.01 = 48.47 P M # fee incurred (48.47 - 2.00) * 1.01 = 46.93 P M # fee incurred (46.93-2.00) * 1.01 = 45.37 P D 1 500 # 50.37 P M # 50.37 * 1.01 = 50.87 P	
Test 3: 1 Checking Account (4 pts) C 1 10000 0.01 5000 200 # 1/2 pt per P P W 1 5001 # fee incurred 49.99 -2.00 = 47.99 P M # 47.99 * 1.01 = 48.46 P D 1 1 # fee incurred 48.46 + 0.01 - 2.00 = 46.47 P M # 46.47 * 1.01 = 46.93 P M # 46.93 * 1.01 = 47.39 P	

D 1 500 # 52.39 P M # 52.39 * 1.01 = 52.91 P	
Test 4: Many Accounts, Many Transactions (4 pts) Posted online	
Test 4: VLD returns no leaks (2 pts)	
Style/Coding Standards (~20%) 9 pts	
Constants are used appropriately. (1 pts)	
Formatting of code: braces, indentations are correct (1 pts)	
Tabs set to 2 (1 pts)	
No Line Wraps (1pts)	
File Header Comments (1 pts)	
Function Header Comments (2 pts)	
Well named variables (2 pts)	

Design (~40%) 18 pts	
UML Document, Wednesday, Feb 20 (1 pts)	
UML Document, Friday, March 1 (2 pts) FINAL DESIGN	
UML Document, Friday, Feb 22 (15 pts) **Specifics** • Diagram Syntax (1 pt) • Accounts hierarchy (5 pts) ◦ balance, interest, account # in parent ◦ fees, minbalance in children ◦ bool flag in SavingsAccount ◦ deposit(), withdraw(), monthlyUpdate virtual in parent ◦ display()-type method virtual in parent ◦ operator>> used correctly • Bank (or something) contains Accounts pointers array (100) (1 pt) • *AccountContainer (if any) does not contain Bank logic (-1)* • Bank (or something) provides necessary interface to perform work (2 pt) • File reading (commands and accounts) encapsulated from Bank (2 pts) • Driver contains little logic, mainly hooking up objects. (1 pts) • Parent class has complete interface for children (1 pt) **General Design (2 pts)** • Single purpose • Open/Closed principal • Liskov substitution principal • Interface segregation • Dependency Inversion	