Item	Possible Points	Earned Points	Notes
Proof • Includes formulas • Shows all test data	**3**	3	Okay
Screen captures	**2**	2	Okay
Program design for Project • The Depositable interface includes the `deposit` method as specified. • The Withdrawable interface includes the `withdraw` method as specified. • The Balanceable interface includes the `getBalance` method as specified. • The Account class: o stores a balance. o has a constructor that takes a number as an argument and assigns the number to the balance. o has get and set methods for the balance. • The CheckingAccount class: o inherits the Account class. o stores a monthly fee. o has a constructor that takes a number as an argument and assigns the number to the monthly fee. o has get and set methods for the monthly fee. o has a method that subtracts the monthly fee from the account balance. o has a method • The SavingsAccount class: o inherits the Account class. o stores a monthly interest rate and a monthly interest payment. o has a constructor that takes a number as an argument and assigns the number to the monthly fee. o has get and set methods for the monthly fee. • The Transactions class contains the two static methods specified. • User input is validated as specified using the MyValidator class • Displays appropriate error messages for invalid data • The results are formatted correctly	**5**	4	The constructor for the CheckingAccount sets the balance for the account to the monthly fee. The constructor for the SavingsAccount sets the balance for the account to the monthly interest rate. This is incorrect. I understand you did this because the Account class has a constructor that requires a parameter. The way to fix this is to add another constructor to the Account class; one that does not require a parameter.

Item	Possible Points	Earned Points	Notes
Design Diagrams: • A correct class diagram is provided for all classes • Design documentation reflects actual logic of code • All methods are documented (one diagram for each method; you may have more than one diagram on a page) • No diagram is larger than one page (8 ½ by 11 inches with ½ inch margins on all sides) • If using flowcharts to diagram the logic: o Each flowchart begins and ends with a terminator symbol Note: the main method beginning terminator contains the word `main()`. The main method ending terminator contains the word `return`. Because you do not write the code that calls the main method, you will not have any flowcharts where the beginning terminator contains the word `START` and the ending terminator contains the word `END`. o The appropriate symbol is used o Only one task per process symbol (the rectangle); each variable declaration should be in its own symbol; show the entire formula for calculations o Every symbol (except a terminator) has at least one flowline leading to it and one and only one flowline leading from it. • If using structured pseudocode to diagram the logic: o The pseudocode is appropriately indented o Each variable declaration is on its own line o The entire formula is shown for calculations o Selection and iteration blocks have a clear beginning and ending • If using Warnier Diagrams to diagram the logic: o Braces are appropriately labeled o Each variable declaration is on its own line o The entire formula is shown for calculations	**5**	3	Severa diagrams are not complete. They have symbols that do not have a flowline leading from them. Some also do not have an ending terminator.
Following course standards: • Code standards: o Code restricted to 80 columns o Follows naming conventions for classes, variables, methods, and constants o Appropriate comment block at top of program file (may use javadoc conventions) o Methods appropriately commented (may use javadoc conventions) o Variables have meaningful names o Braces align correctly o Control statements formatted correctly • All non-code files contain your name, the course code (CITP 190), and the project number at the top of the file. • All design diagrams are in one file. • All files are in standard 8 ½ by 11 inch format with at least ½ inch margins on all sides of the page.	**5**	5	Nice comment blocks.
Penalties: • Incorrect calculations • Output is not presented as shown (including spelling and spacing) • Code does not follow the standards • Not all test data was used • Reflects material outside what has been covered through Chapter 8 • Using any classes not mentioned in the instructions or does not use one of the classes mentioned in the instructions • Using a `continue` statement or misusing a `break` statement	**-20 for any of the items listed**		
Total	**20**	**17.0**	