

CPSC 233 – Team Assignment 5 – Requirements

As a team, complete a solution for the individual assignment that you'll use for your team submission. Add the code that would react appropriately to button clicks by:

1. Add an instance variable of type `BankAccount` to `BankApp`.
2. Make the widget where the user enters the amount to deposit/withdraw an instance variable.
3. Make the label that displays the account balance an instance variable as well.
4. Create an inner class that implements `EventHandler<ActionEvent>` that handles deposits by
 - a. Implementing the `handle` method such that it gets the amount entered in the `TextField` (using the instance variable), and
 - b. Uses the `BankAccount` instance variable to deposit the specified amount into the account, and
 - c. Update the balance label to reflect the new balance in the account by calling the `getBalance` method on the `BankAccount` instance variable.
 - d. Adding an instance of this event handler to the deposit button through its `setOnAction` method.
5. Create an inner class/anonymous inner class that implements `EventHandler<ActionEvent>` that handles withdrawals using similar steps to the deposit described in the previous point.

Rubric

	2 marks	1 mark	0 marks
Interface	The interface contains the required buttons, textfield and labels. The widgets are well laid out in the window.	The widgets look awkward in their layout or the widgets have a fixed place so resizing the window makes it look odd.	The interface does not have the required widgets.
Event Handling	The deposit and withdraw buttons behave as expected with all account information display updated.	Only deposit or withdraw is working.	Buttons don't seem to do anything or are not working correctly.
Legibility and Documentation	Code is easy to read. Good use of white space and good variable names. Code is never indented more than three deep (inside a method). Good use of javadoc and in-line documentation.	Code is easy to read but documentation needs to be updated. OR Good documentation but code legibility needs to be improved.	Code is difficult to read or documentation is missing.
Code Design	Classes are well designed. Good use of inner classes, or separate classes to handle events.	Class is mostly well designed but there is some improvement possible.	Some classes needed to be redesigned.