I annotated this because I thought I could improve a bit on the implementation details for the app. My implementation turned out a bit different than you might have expected, but all the required elements are present, as is the specified functionality. I just wanted to point that out.

Thanks

Rob

CIS162AD Rio Salado College C#: Level I

Final Project Instructions

Final Project

For your Final Project, you will create a banking account register. The register will include Account Name, Account Number, and Account Balance, and it will allow a user to enter deposit or withdrawal information. Throughout the chapters, you were given all of the tools necessary to complete this task. Use a flowchart to help with your decisions and the programming example in Chapter 9 to help with the project. Good luck!

Rules

1. The initial graphical user interface should contain a label that contains your name and MEID#, and it should provide two user interaction options: Enter Account Information and Clear.

The form contains name & MEID. These are always visible.

The **initial state** of the form is to enter Account Information-- I felt that Clear was not really valid until they had something *to* clear. The user has an obvious choice, to enter the account details, and proceed with the entering of transactions.

Once the user has entered the state where transactions are entered, then the Clear button is available. Hitting the button at that point will exit the transaction state and return to the Account Information state.

2. When the user selects Enter Account Information, the graphical user interface should provide three options: Account Name, Account Number, and Beginning Balance.

The Account Information state provides the user with the ability to enter these fields.

3. After the user enters the correct information, allow him/her to continue by using the Continue Option.

The Continue button is disabled until the user successfully passes validation on all 3 fields.

4. The selection of Continue should provide the user with Withdrawal Amount and Deposit Amount.

When the user hits Continue, the lower group box is enabled. This is where the user performs activities on the account (depositing and withdrawing).

5. The Continue button selection should also make the label Beginning Balance read Available Balance, and the text boxes in Account Information should become read-only.

This is what happens when Continue is hit.

6. Allow users to enter Withdrawal Amount and Deposit Amount and change Account Balance accordingly.

They can do this when in the Activity group box.

7. Clear Withdrawal and Deposit amounts/text boxes so users can continue to enter additional amounts.

This happens.

8. The user should be able to add as many withdrawals and deposits as he/she wants.

They can, as long as they stay within the rules.

9. The Clear button should clear the program and return to Step 1 whenever it is selected.

The Clear button exits the Activity state and disables the group box. It enables the Account Details group box, and all fields and controls within.

At this point the user can either hit Continue again, which takes them back into the Activity group box, or they can change any of the three fields, or they can hit the “clear input” link label to completely clear everything, at which point they must re-enter all the data and pass validation to enable the Continue button once again.

I did it this way so the account details won’t have to be re-entered. This way the user can easily reset the starting balance if they like, and re-enter the activity state. If they do want to completerly clear everything out they can do that using the link label.

Design Rules

1. Completing the flowchart is required. If you insist
2. The flowchart must contain detailed process flow, defined variables, and decisions.

Validation

1. All TextBoxes that contain a numeric value should be validated. Done.
2. If an incorrect value is entered, notify the user and allow the user to retry. Done.

3. Negative values are allowed. Done.

CIS162AD Rio Salado College C#: Level I

Objects and Structures

1. Separate the business logic from the presentation layer in the form of a class. You performed this task in Chapter 4.

I created a BankRegister class. This class encapsulates all logic related to maintaining the balance using transactions.

2. Use visible property to activate fields with a button click. (This is explained on pages 490 − 492
of the textbook and can be found in Chapter 9's programming example: “TempAgency.”)

I am using a button click event to make the “clear input” link label invisible when the Continue button is hit.

If the point is to show that I understand how to change property values within event handlers, I am also enabling/disabling controls inside event handlers, as well as setting focus.

3. The Clear method can be found on pages 371, 428, 438 − 440, 492, and 496 of the textbook.

OK I used it.

4. Follow the rules stated above to create the program.

Form Layout

1. The form should include a label with your name and MEID# that is visible at all times.
It does

2. The user should be able to click a button to enter account information or clear the form.
When in the activity state, The user can click the Clear button to return to the Account Details group box, where they can enter data.

3. Your form should include the Account Name, Account Number, and Beginning Balance.
It does.

4. The user should be able to click a button to enter withdrawal and deposit amounts.
They can.

5. The user should be able to click a button to submit withdrawal and deposit amounts.
They can.

6. The user should be able to clear the form by pressing the Clear button.
They can return to the Account Details group box by hitting the Clear button. They can partially or completely clear the the form at that point by manually clearing fields or by hitting the “clear input” link label.