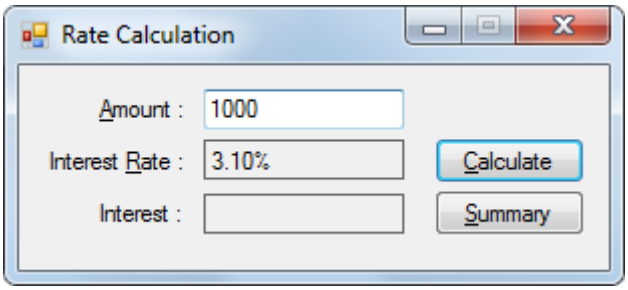
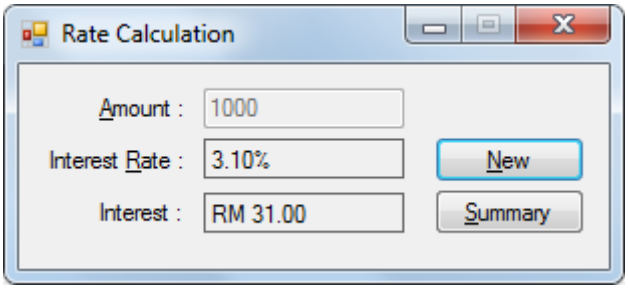
Practical 6: Multiple Forms

Q1.

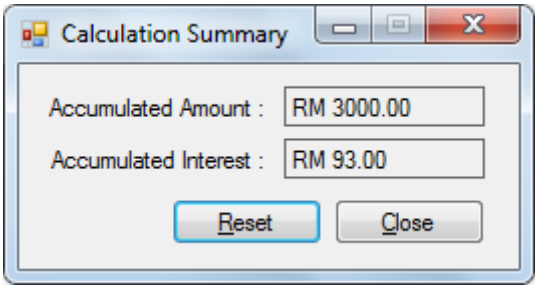
FrmSplash: Display during project startup.



FrmCalculate (before): Click [Calculate] to calculate interest. Click [Summary] to load **FrmSummary** as modal dialog.



FrmCalculate (after): Change button's text to [New] and disable [Amount] textbox after calculation.



FrmSummary: Show accumulated amount and interest.

Create a VB Windows Form Application with the following 3 forms:

(a) FrmSplash

- Customize the form with your preferable layout. Set the form's **FormBorderStyle** property "None" (to make it a borderless form).
- Set the form as the splash screen for the application.

(b) FrmCalculate

- Declare an interest rate of 3.10% as a constant. Display the rate during form **Load** event.
- Declare public variables for holding the values for accumulated amount and interest.
- Create a sub procedure named **DoCalculate()** to calculate and display the interest. After that, change the [Calculate] button's **Text** property to "New" (do not forget the access key if any), and disable the [Amount] textbox. In addition, increase the accumulated amount and interest accordingly.

```
Interest = Amount * Interest Rate
```

NOTE: The button will switch its **Text** between "Calculate" and "New". Do not create 2 separated buttons for [Calculate] and [New]. Use a single button and modify its **Text** accordingly during run-time.

- Create another sub procedure named **DoNew()** to change the [New] button's Text property back to "Calculate" (do not forget the access key if any), and empty [Amount] textbox and [Interest] label. Finally, enable and focus on [Amount] textbox.

- Handle the **Click** event for [Calculate] button. If the button's Text is "Calculate" (do not forget the access key if any), call the **DoCalculate()** sub procedure. Otherwise, call the **DoNew()** sub procedure.

- Handle the **Click** event for [Summary] button. Show **FrmSummary** as a modal dialog.

(c) FrmSummary

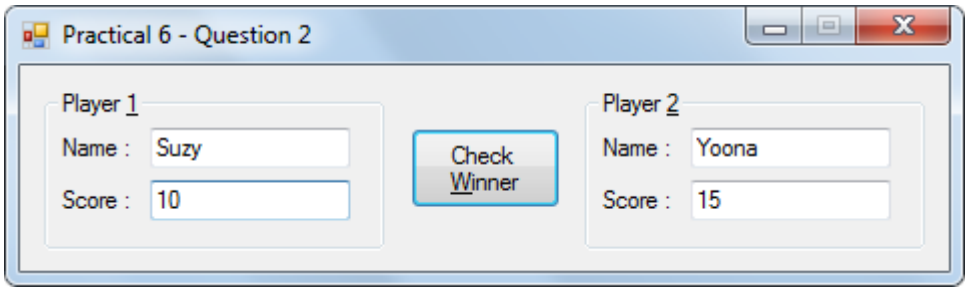
- Display the accumulated amount and interest during the form **Load** event.

  NOTE: Doing so by accessing the public variables at **FrmCalculate**.

- Handle the **Click** event for [Reset] button. Reset the accumulated amount and interest back to zero.

- Handle the **Click** event for [Close] button. Close the form.

Q2.



FrmCheck: Allow user to enter name and score for player 1 and 2, then check who is the winner.

FrmWinner (winning): Display the name and score of the winning player.

Create another VB Windows Form Application with the following 2 forms:

(a) FrmCheck

- Handle the **Click** event for [Check Winner] button. Compare the scores (Integers) from both players, determine the winner, and finally show **FrmWinner** as a modal dialog:

- If there is a winner, display the winner's name and score at the respective labels in **FrmWinner**.

- If it is a draw game, display "N/A" at the respective labels in **FrmWinner** (as shown below).

FrmWinner (graw game): Display "N/A" at the respective labels.

NOTE: The logic of determining the winner is to be programmed in **FrmCheck**. You can access to the labels in **FrmWinner** by using the following method:

```
FrmWinner.lblTitle.Text = "The Winner Is"
FrmWinner.lblName.Text = n1
FrmWinner.lblScore.Text = CStr(s1)
```

Or, using the **WITH** block:

```
With FrmWinner
    .lblTitle.Text = "The Winner Is"
    .lblName.Text = n1
    .lblScore.Text = CStr(s1)
End With
```

Remember to handle possible input errors too.

(b) FrmWinner

- Handle the **Click** event for [Close] button. Close the form.