Loops in Visual Basic II

IMPORTANT: While writing programs involving loops, sometimes you may accidentally write code that contains an *infinite loop*. An infinite loop will cause your display to freeze up or output continually to a text box, depending on what your loop contains. If this happens you can break out of the infinite loop and stop the run of the program by clicking the Stop icon (a red square to the right of the Play icon).

Topics

- count-controlled loops based on a computed value
- loops whose condition depends on an event and not a count
- integer division
- compound conditions
- dealing with error cases resulting from bad user input (division be zero)

Preliminaries

Go to today's lab folder (`Lab05_Loops`) on the `N:` drive and copy the write-up for today (**not the whole folder**), `Lab-II.pdf`, to your `M:\CS130\Labs\Lab05_YourLastName_YourFirstName` folder. You may now work locally by opening the write-up from within the copied folder.

Interest calculator

4. Launch the RAPTOR software and use it to create and save a new program called `InterestCalculator_Part4.rap` inside of your lab folder.

   Complete `InterestCalculator_Part4.rap` so that, given a starting principal and a nominal interest, it computes the number of years of compounding interest required to reach a target principal. Assume the interest is compounded annually and that no withdrawals are made from the account for the entire period.

   Your program should produce one output which is a statement containing the total amount of interest earned in a statement in the following form:

   ```
   After 7 years of compounding, your savings account with an initial balance
   of $500 and APR of 3%, will have reached your target balance of $600 and have
   a current balance of $614.9369.
   ```

   After you get your program working, fill out `Test Table 4` in the file `InterestCalculator_TestTables.docx`, making sure that your expected output and actual output are consistent.

When done, save and close file `InterestCalculator_Part4.rap`.

Checkpoint 5 (80/100): A successful `InterestCalculator_Part4.rap` program:

- uses a loop statement
- produces correct output for all test cases in Test Table 4
- produces correctly formatted output according to the example above
- must have successfully completed Checkpoint 4

5. Use **File Explorer** to navigate to your lab folder. Copy and paste the folder `InterestCalculator_Part3`, and then right-click on the copy to rename it as `InterestCalculator_Part5`. Launch VS Express and use it to open the project `InterestCalculator_Part5`.

   Complete the action for `btnComputeNumberOfYears` so that it is equivalent to your RAPTOR program. You MUST use a `while`-loop.

   As with Part 3, you MUST NOT change the design of the form. If you need additional input, besides the two text boxes, then you should use and `InputBox`.

   Output for the Visual Basic version of this program should use the `FormatCurrency` and `FormatPercent` functions to produce output that has the following format:

   ```
   After 7 years of compounding, your savings account with an initial balance
   of $500.00 and APR of 3.00%, will have reached your target balance of $600.00
   and have a current balance of $614.94.
   ```

   After you get your project working, fill out `Test Table 5` in the file `InterestCalculator_TestTables.docx`, making sure that your expected output and actual output are consistent.

Your project should behave identically to the solution which can be run by double-clicking the file `InterestCalculator_Part5.exe` found in the `Executables` folder inside of your lab folder.

When done, save and close project `InterestCalculator_Part5`.

Checkpoint 6 (85/100): A successful `InterestCalculator_Part5` project:

- is based on a correct `InterestCalculator_Part4.rap` RAPTOR program
- produces correct output for all test cases in Test Table 5
- produces correctly formatted output according to the example above
- must have successfully completed Checkpoint 5

6. Use **File Explorer** to navigate to your lab folder. Copy and paste the folder `InterestCalculator_Part5`, and then right-click on the copy to rename it as `InterestCalculator_Part6`. Launch VS Express and use it to open the project `InterestCalculator_Part6`.

   Improve the action for `btnComputeNumberOfYears` so that in the case that the user enters a target balance that it less than or equal to the starting balance they are prompted to enter another value. This should be repeated until the user enters a target balance that is greater than the starting balance.

   After you get your project working, fill out `Test Table 6` in the file `InterestCalculator_TestTables.docx`, making sure that your expected output and actual output are consistent.

Your project should behave identically to the solution which can be run by double-clicking the file `InterestCalculator_Part6.exe` found in the `Executables` folder inside of your lab folder.

When done, save and close project `InterestCalculator_Part6`.

Checkpoint 7 (95/100): A successful `InterestCalculator_Part6` project:

- re-prompts user for input when invalid values are input
- produces correct output for all test cases in Test Table 6
- produces correctly formatted output according to the example from Part 5
- must have successfully completed Checkpoint 6

7. Use **File Explorer** to navigate to your lab folder. Copy and paste the folder `InterestCalculator_Part6`, and then right-click on the copy to rename it as `InterestCalculator_Part7`. Launch VS Express and use it to open the project `InterestCalculator_Part7`.

   Improve the action for `btnComputeNumberOfYears` so that instead of outputting the statement from Part5, a table of interest and balance is output. This requires no additional calculation, only outputting the intermediate values that are being computed. The table produced should have the following format:

```
Starting balance: $500.00
APR: 3.00%
Target balance: $600.00
**************************

Year    Interest       Balance
============================
1         $15.00       $515.00
2         $15.45       $530.45
3         $15.91       $546.36
4         $16.39       $562.75
5         $16.88       $579.64
6         $17.39       $597.03
7         $17.91       $614.94
============================
         $114.94       $614.94
```

Your project should behave identically to the solution which can be run by double-clicking the file `InterestCalculator_Part7.exe` found in the `Executables` folder inside of your lab folder.

When done, save and close project `InterestCalculator_Part7`.

Checkpoint 8 (100/100): A successful `InterestCalculator_Part7` project:

- produces correctly formatted output according to the example above
- produces correct output for all test cases in Test Table 7
- must have successfully completed Checkpoint 7

Submission Instructions

Your `M:\CS130\Labs\Lab05_YourLastName_YourFirstName` folder should contain your solutions to this and Tuesday's lab.

To submit your work, copy this folder and paste it to `N:/Handins/CS130/Lab05_Loops` PRIOR to midnight tonight. Submissions received after 11:59pm tonight will be considered late and will receive a grade of 0.

You are not allowed to seek help from TAs on this lab outside lab time. TAs have been specifically instructed NOT to provide any help so please refrain from this activity.