MIS 316

SPRING 2020

PROJECT 3

Demonstrate Loops / For Statements

1. Use your previous project folder / and solution file and continue to expand your knowledge of C# and loop statements. Open your project (click on the .sln file on your desktop)

2. Fix any errors on the as noted in Entropy for Project 2 and test any changes.

3. Add a new page / and update the menu
    a. In your solution explorer, right click on the name of your project (should be bolded), then
    b. Add > Add New Item > WebForm
    c. Name this new page Project3 (no space)
    d. Click the Select Master Page
    e. Add > Select site.Master
    f. You should now have a new page in your solution explorer: Project3

4. Open the site.Master and modify the black navigation bar to direct users to this page
    a. Open the site.master page
    b. Look around row 55 where you have them directed to Project 2
    c. Copy this row and modify the href portion to direct them to Project2 and change the text to be Project 3 (with space)
    d. Save and Test, when you click Project 3 does it take you to your new page?

5. Close the site.master and have open only the Project3.aspx page

6. Change the title tag in line 1 to be Title=”Your Name – Project 3”

7. Insert a few blank rows between lines 3 and 4; as </asp:Content> needs to be the last line on the page

8. Copy from the Chapter4-Loops page we did in class, all of the tags between the <asp:content and the </asp:content> tags. You will be copying all of the headings, labels/textboxes and then modifying for the new page

9. Modify the heading and paragraph tags at the top of the page:
    a. Your Name (in the h2)
    b. Project 3 (in the h3)
    c. Demonstrate Loop Statements (in the p)
    d. Save

10. **First Button (LOOP).** You will want to know how many years it will take to pay off a mortgage.

    a. After the first , modify the objects as follows.
    b. Modify the H3 tag to state: How many years to save money for a deposit?
    c. Modify the first label to state: Starting Mortgage Amount
    d. Modify the ID of the text box to be __1StartingMortgage
    e. Modify the Control to Validate property to be _1StartingMortgage
    f. Change the Error Message of the validation control
    g. Update the compare validator for the proper messages and control to validate
    h. Rename your second text box (and a property…) to be _Interest Rate, and the label text property to reflect the entry desired
    i. Change both validation controls to reflect the above
    j. Rename your third text box (and a property…) to be _AnnualPayment, and the label text property to reflect the entry desired
    k. Change both validation controls to reflect the above
    l.
    m. Change the button to display: Number of Years Needed

    n. Thus if you have a mortgage of 100,000 and an annual payment of 12,000, and 5% interest, at the end of one year you would owe.
        i. Assume you pay interest on what you owe at the beginning of the year
        ii. End of Year Balance would be: 100,000 + (intRate * 100,000) – 12,000, yielding 93,000
    o. On the aspx page, look for any onClick=…., and delete the instructions for the onClick=…… (you should delete the onClick as well as the code in "" after the =).
    p. After the while loop, do a count and display the number of years in _1Answer
    q. Run your project and see if the first entry renders properly

11. **Second Button** (**WHILE**) – This button should show all the squares from 1 to 10 in the list box. You will not need any input items, just the button
    a. Thus copy the <!—Group C to the end of Group C → to your page
    b. ***Delete all items*** before the Button
    c. Update remaining items in this group:
        i. Rename btn to be _2Calculate
        ii. Change the validation group
        iii. Delete the onClick
        iv. Rename the 2nd label to be _2Answer
    d. Double Click the button, and add the instruction to clear the list box and then the remainder of the steps to create the squares from 1 to 10. You must use a While statement here, using a intCounter to tell you when to stop, You will not have a grab values from text boxes section here
    e. Your output in the list box should be
        1 – 1
        2 – 4
        3 – 9…. (showing the number and the number squared)

12. **Third Button** (**FOR**) – This button should show all the squares from 1 to 10 in the list box. You will not need any input items
    a. Copy the 2nd group of tasks
    b. Rename all objects, validation groups, and text properties
    c. Add the instruction to clear the list box as the first instruction in the calculate section
    d. Your output in the list box should be
       1 – 1
       2 – 4
       3 – 9…. (showing the number and the number squared)
    e. You must use a **FOR** statement, as well as declare the starting and ending variables for the FOR statement.

13. **Fourth Button** (FOR with IF) – Your goal will be to decide if the user has entered a word with the letter E in the text box and count the number of E's in the text box.
    a. Copy the 1st group tasks and modify as necessary. Delete the 2nd and 3rd labels/textboxes and validation ccontrol
    b. This area should have 1 text box, 1 label for the text box, 1 required field validator for the text box (delete the 1st compare validator), 1 button, and the __4Answer label.
    c. Rename as necessary, change the text property where needed
    d. Change the validation groups
    e. Instruct the user to enter any word into the text box
    f. You should check to insure something has been entered into the text box

    g. Hints:
        i. Declare variables (one for the textbox in, one for the length of the string, one for the counter, one for the # of E's)
        ii. Once you grab the value from the text box, you will want to convert it to UPPERCASE
            strWordIn = ___4WordIn.Text.ToUpper();
        iii. You will then want to determine how many letters are in the string
            intLength = strXxxxx.Length
        iv. Write the **For** statement that will increment the intCounter from 1 to the variable that contains the length of the string
        v. Inside the FOR statement do an IF statement to determine if the letter E is found in the word (you will check each position looking for the E)
        vi. Help:
            1. Your IF statement should check the substring, one character at a time as in :
            2. if (strWordIn.Substring(intCounter,1) == "E")
                a. thus if the above is true, add one to the E counter!

14. Test your work, once you are sure your project is working,
    a. Close Visual Studio
    b. Copy your ENTIRE folder from your desktop (do not drill down and copy the inside items), copy the entire folder on your desktop to your miscapstone server area
    c. Do NOT place in the GRADED folder found in your miscapstone area