

MIS 316  
SPRING 2020  
PROJECT 4

Use of Methods and Reusable Code

1. Help? I will have virtual office hours most Monday to Thursday from 9 am to Noon. The link to click to get to my 'zoom' room is on the class web site. You might want to send me an email if I don't respond when you get to the zoom room.
2. Scanning Chapter 5 of your text will help you understand what we are attempting to do for this project.
3. New ACCESS to your projects using HORIZON. I recommend Horizon as it will avoid the VPN issues (as the school only has a limited number of concurrent VPN licenses).
4. See the class website for the PDF on how to use HORIZON.
5. I have modified all of your miscapstone folders to have your basic files (project3.aspx, project2.aspx at root level in your project. This will let you use Horizon and when using Visual Studio, you will now FILE OPEN WEBSITE versus a project (see PDF in #3 above)
6. Please note I have copied a Chapter5 sample files and your starting files for Project4 in your project. This should help you start a little easier.
7. Fix any errors on the as noted in Entropy for Projects 2/3 and test any changes.
8. Open the Chapter 5 files, run it and look at the code that will 'call' one method (the CountLettersInString). It will call the method 5 times, once for each vowel. This method saves us from having to write the loop and if statement five times.
9. Open the Project4 aspx starting file
  - a. Update the title tag – first line
  - b. Update the H2 tag with your name
  - c. Save
10. 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 56 where you have them directed to Project 3
  - c. Copy this row and modify the href portion to direct them to Project4 and change the text to be Project 4 (with space)
  - d. Save and Test, when you click Project 4 does it take you to your new page?
11. Close the site.master and have open only the Project4.aspx page

12. **Product 1 Sales** – your task is to calculate the freight charge, and the sales tax and then the total using methods to calculate the freight charge and sales tax. We will want to use the same formulas under more than one calculate button so it makes sense to build a reusable method.

13. Under the \_1Calculate button:

- a. Clear the list box
- b. Also modify the lstLabelText to read: Product 1 Information

14. On the C# side of Project4:

- a. Before the last } build a Method (name it freightCharge), to help you get started: 

```
public double freightCharge(double dblSalesIn)
{
    double dblFreightCalculation = 0;
```
- b. The freightCharge method should accept one double variable and return a double with the amount of the freight (done above)
- c. The freight charge should be calculated as (build a series of if's/else if's):
  - i. Sales <= 25, freight charge 10%
  - ii. Sales 25.01 to 50.00, freight charge 8%
  - iii. Sales 50.01 to 100.00, freight charge 6%
  - iv. Sales over 100.01, free
- d. Return dblFreightCalculation

15. Create a second Method (name it salesTax), it should look very similar to the above.

- a. It should accept a double in (the Sale Amount from the text box + the freight previously calculated)
- b. Sales tax should be calculated as 7% of the Sale Amount + Freight
- c. Return the sales tax

16. Under the \_1Calculate Button

- a. Declare your variables, recommend one for the text box, one for sales amount, one for freight, one for sales tax and one for grand total.
- b. Your sales amount should equal the amount in the text box, show this amount in the list box (Purchased Amount = . \$....
- c. Call the freight method and add to the grand total , show the amount of freight in the list box (Freight = \$
- d. Call the sales tax method, (it should pass the total of Sale Amount and freight or grand total) show the amount of sales tax in the list box, add sales tax to your grand total, (Sales Tax = \$
- e. Finally, show the grand total in the list box (Total = \$

17. Example:

- a. Sales of 30.00 would yield, Purchase Amount = \$30.00, Freight of \$2.40, sales tax \$2.27 and a grand total of \$34.67

18. **Product 2 Sales** –This button should follow the same process as button 1

- a. Clear the list box
- b. Change the text of the lstLabelText to be Project 2 Sales
- c. Under your \_2Calculate button, it should follow the same process as you did under \_1Calculate, you do not need to build new methods.

19. **Calculate Grade Point Average** (3<sup>rd</sup> button)

- a. Copy the 1<sup>st</sup> series of label/text box / error checking and place 4 more text boxes to capture the grades for one student for one semester. Hint: Start by copying lines 61 to 69 and pasting 4 more times
- b. Rename the 2<sup>nd</sup> to 5<sup>th</sup> objects as \_3Grade2, \_3Grade3 etc
- c. You should only have one calculate button which is already in the project

20. Create a method to return the Grade Point Average, it should accept a string In (the actual grade, and return a double reflecting the grade points earned for a 3 credit course. For example, for an A return 12, for a B return 9, etc. We will not consider + and –'s in this exercise. Recommend you name your method:  
public double GPACalculator (string strGradeIn)

21. Under your \_3Calculate button, grab the 5 values, convert to upper case (see the sample chapter 5 pages provided to you)

- a. Make sure to clear the list box
- b. Call the method above 5 times (once for each text box
- c. Add the points earned from each method call to a grandTotal
- d. After the five calls, divide the grand total / 15 and show that GPA average in the list box

22. Test your work, once you are sure your project is working,

- a. Close Visual Studio
- b. If you are using Horizon then your files will already be on miscapstone (saving you from having to VPN from home)
- c. But to enable grading, please send me an email with a subject line:  
MIS 316 Project 4 Ready

23. Again I do have virtual office hours should you need help, so don't hesitate to use zoom to find me in the mornings.