|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CS1400 Lab #12**  **Designing the Sales Invoice Class**  **Version 1.0**  **Introduction**  In this lab, you will design a class that represents a sales invoice. Later you will write the Windows Forms Application that uses this ***UML Class Diagram*** in this lab. Your Sales Invoice will only consider the simple case where someone buys a given quantity at a unit price for a single item, for example, three boxes of nails at $12.50 each. The user interface for your program could look something like the Form in Fig. 1.    Figure 1 Sales Invoice Calculator Start Form  The user enters in the number of items purchased and the cost per item. Your program will produce an invoice that shows the total cost for these items, including state and local sales taxes and total amount due. An example of how this might look is shown below in Fig. 2:    Figure 2 Sales Invoice for 12 Units @ $1.25  Design a ***SalesInvoice*** ***class*** that contains all of the data that you think your program needs to keep track of, and ***ALL*** of the business logic required to produce a sales invoice as shown. Let the ***state sales tax*** rate be ***6.7256%*** and the ***local sales tax*** rate be ***2.5666%.*** Create a ***UML Class Diagram*** for this class. You can create the UML Class Diagram in any application you would like, such as Paint, PowerPoint, Word, pdf, etc. It cannot be handwritten!  **File(s) to Submit:**  Place your UML Class Diagram in a PDF file and zip the file. Name the zip file Lab\_12\_your-initials\_V1.0.zip. For example, I would name my file Lab\_12\_DAF\_V1.0.zip. Submit this assignment as Lab #12 on Canvas. Do not submit any other files.   |  |  |  | | --- | --- | --- | |  | **Grading Checklist** |  | | # | Program | C(correct)  X(incorrect) | | 1 | Meets & works to specifications | 6 points | | 2 | Error Free, elegant & efficient | 4 points | | 3 | Pseudo-Code | -3 points | | 4 | Style Guidelines | -2 points | | 6 | Source Files(s) & Formatting | -2 points | | 7 | Project Prolog | -1 points | | 8 | Function Prologs | -1 points | | 9 | Zip Filename | -1 points | | 10 | Lab & Project Names | -1 points | | 11 | Zip File is invalid or will not unzip | Lab = 0 pts | |  | Total Points | 10 | 0-9 | |