CIS 605 – Fall 2024
Assignment Set 7
Due Date: Friday, December 6 @ 11:59 PM

Develop the project described below using good visual design and program coding practices that include

- Professional Appearance (Layout, placement, spelling, formatting)
- Meaningful title on title bar of form(s)
- Identifiers (names) for objects, variables, and constants are meaningful and follow a consistent naming convention
- General remarks at the start of every class in your program including Class Name, Class Description, Developer Name, Date Created, Date Last Modified
- Descriptive remarks for every method
- Proper indentation & blank line after each full comment line
- All variables and constants are local whenever possible (scope)
- Modular programming – i.e., breaking down a "large" programming task into multiple, independent modules, with each module performing one part of the required functionality.

Caution: If the business logic is contained in the form class, you will receive zero credit.

Program 16

1 Enumeration (named, CrustAndSize) for Pizza Type – Crunchy Thin 10", Deep Dish 11", Hand Tossed 12", Gluten Free 13", Italiano 14", NY Style 15", Big Kahuna 16" (**note:** place this enumeration in the Order.cs file, but outside the class block

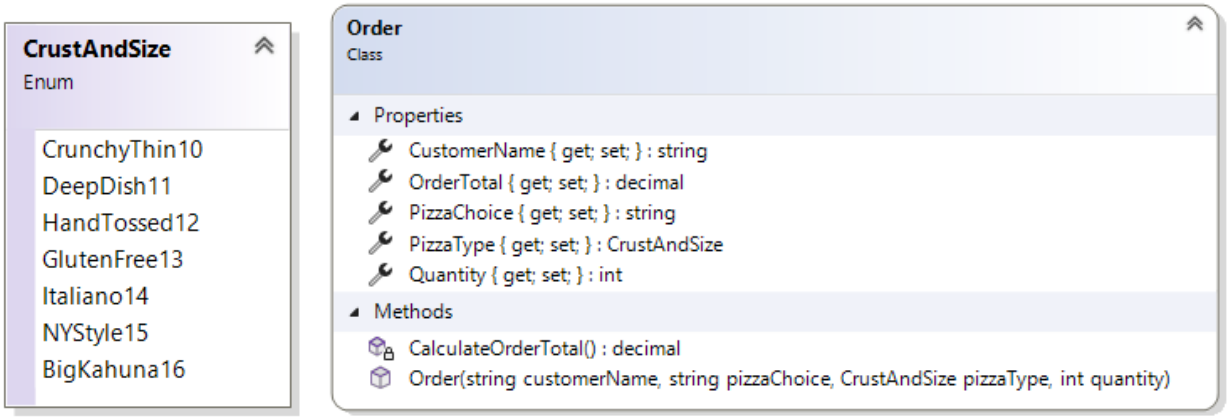
Create an Order class that has:

- 5 instance properties (auto-implemented – public get and private set)
  - Customer Name
  - Pizza Choice
  - Pizza Type
  - Quantity
  - Order Total

- 1 constructor that
  - instantiates object and set its customer name, pizza choice, pizza type, and quantity using parameters
  - calls a private method (see below) to set the order total.

- 1 method
  - a private method to calculate and return the order total based on the pricing table below:

Pizza Type	Price
Crunchy Thin 10"	$15.99
Deep Dish 11"	$16.89
Hand Tossed 12"	$17.79
Gluten Free 13"	$18.69
Italiano 14"	$19.59
NY Style 15"	$20.49
Big Kahuna 16"	$21.39

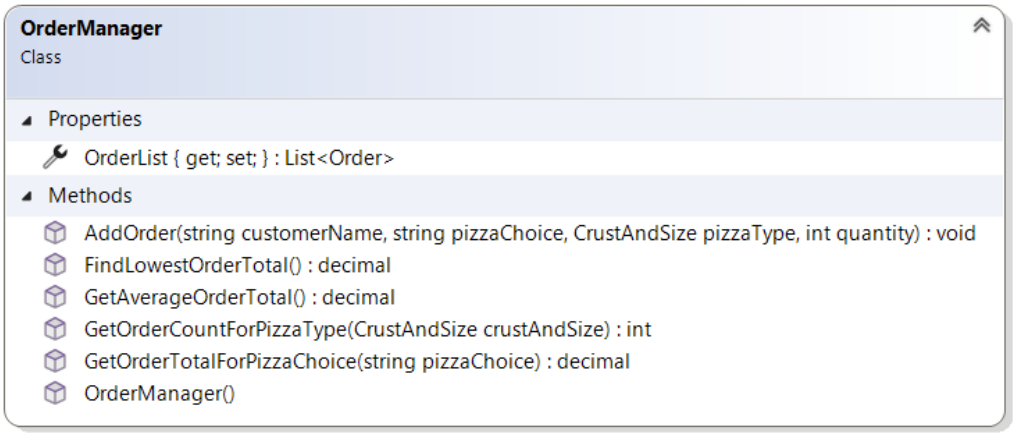


Create a OrderManager class that has:

- 1 instance property (auto-implemented – public get and private set)
  - A List <Order> collection to hold Order objects

- 1 constructor that
  - instantiates object and instantiates the List <Order> collection

- 5 public methods
  - a method to a) instantiate an Order object and add it to the List <Order> collection
  - a method to find and return the lowest order total among all the orders
  - a method to sum up and return the total order amount of all orders for a specific pizza choice
  - a method to calculate and return the average order total
  - a method to count and return the number of orders for a specific pizza type



Create a form class that has the following controls and functionality

- A Menu:

Display
All Orders
Total Order Amount for a Pizza Choice
Number of Orders for a Pizza Type
Orders Summary

_____

Exit

Edit
Reset for Next Order

_____

Add Pizza Choice
Remove Pizza Choice

    - Disable the menu items (with the exception of *Exit*) in the *Display* menu. Enable them after the *Add Order* button (see below) is clicked.

- Appropriate controls for entering the customer's name and quantity

- A drop-down combo box that contains the pizza choices: Jalapeno Popper, Sweet n Spicy, Krazy Hawaiian, Buffalo Chicken, Double Pepperoni, Mile High, South of Border.

- A drop-down list box that displays the pizza types: Crunchy Thin 10", Deep Dish 11", Hand Tossed 12", Gluten Free 13", Italiano 14", NY Style 15", Big Kahuna 16". Populate this list box using the CrustAndSize enumeration (hint: refer to Module8Ex2).

- An *Add Order* button that

    - validates the input data (e.g., customer's name is entered, pizza choice and pizza type are selected),
    - instantiates a OrderManager object (once; when the first order is added),
    - calls the appropriate method on the OrderManager object to add the order
    - enables the four menu items within the *Display* menu

- The *All Orders* menu item should display (**in another form**) all the orders in a DataGridView. Display your name and an appropriate title at the top of the form.

- The *Total Order Amount for a Pizza Choice* menu item should display (**in a message box**) the total order amount of all orders for a specific pizza choice selected by the user from the drop-down combo box. Make sure the user has selected a pizza choice before calling the method in the OrderManger class.

- The *Number of Orders for a Pizza Type* menu item should display (**in a message box**) the number of orders for a specific pizza type selected by the user from the drop-down combo box. Make sure the user has selected a pizza type before calling the method in the OrderManger class.

- The *Orders Summary* menu item should display (**in another form**) the lowest order total, and the average order total of all the orders. Display your name and an appropriate title at the top of the form.

- The *Add Pizza Choice* menu item should add to the items collection of the combo box a new pizza choice that is entered into the text box (of the combo box). Make sure the pizza choice to be added does not already exist in the items collection. If it does, inform the user and do not add the duplicate item (hint: refer to Modeul5Ex3).

- The *Remove Pizza Choice* menu item should remove the selected pizza choice from the items collection of the combo box.

- The *Reset for Next Order* menu item should reset the input controls.

- Attach a context menu to the combo box with the following items: *Add Pizza Choice and Remove Pizza Choice.*