CIS 605 – Fall 2024
Assignment Set 4
Due Date: October 13 @ 11:59 PM

Develop the projects described below using good visual design and program coding practices that includes

- 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.

Please Note: If any calculations are performed in the form class, you will receive zero credit for that program.

Program 9

Create a project to calculate conference charges for CSU Conference Services. The program must determine the charges for a conference based on accommodation type, number of attendees, number of nights of stay and optional services.

The **Conference** class should have

- 1 Enumeration for Accommodation Type – Single, Double, or Suite (**note:** place this enumeration in the Conference.cs file, but outside the class block

- 10 Instance Properties
    - Name of the conference (auto-implemented – public get and set)
    - Accommodation charge (auto-implemented – public get and private set)
    - Optional services charge (auto-implemented – public get and private set)
        - **note:** optional services are Internet access and recreation center access
    - Discount (auto-implemented – public get and private set)
        - **note:** a discount is provided if the conference has 60+ attendees staying 7+ nights
    - Total charge for the conference (auto-implemented – public get and private set)

    - Number of attendees (with a private field and public get and set accessors)
    - Number of nights (with a private field and public get and set accessors)
    - Accommodation choice (with a private field and public get and set accessors)
    - Internet access (with a private field and public get and set accessors)
    - Recreation center access (with a private field and public get and set accessors)

After setting their respective values, the set accessors for number of attendees, number of nights, accommodation choice, Internet access, and recreation center access should call the instance method that calculates the conference charges (see below).

- 1 Constructor
    - To instantiate object and set six properties – name of the conference, number of attendees, number of nights, accommodation choice, Internet access, and recreation center access

- 1 Private Instance Method
    - To return the rate for the chosen accommodation type: Single = $76.45, Double = $51.93, Suite = $116.72 (**note:** rates are per attendee/per night):
    - Use a switch statement in this method

- 1 Private Instance Method (to calculate conference charges)
    - To calculate and set
        - accommodation charge
        - optional services charge for Internet access and/or recreation center access
        - discount – if there are 60+ attendees <u>and</u> they will be staying 7+ nights
        - total charge for the conference
    - Calculations are as follows:
        - accommodation charge = rate for chosen accommodation type (determined by calling the previous instance method) * number of attendees * number of nights
        - optional services charge = Internet access charge + recreation center access charge
            - Internet access charge = if Internet access is requested, $8.22 * number of attendees * number of nights
            - Recreation center access charge = if recreation center access is requested, $7.74 * number of attendees * number of nights
        - discount = if there are 60+ attendees <u>and</u> they will be staying 7+ nights, 15% of accommodation charge + 8.25% of optional services charge
        - total charge for the conference = accommodation charge + optional services charge - discount

The form class should have appropriate controls for entering information for a conference including a) a text box for the name of the conference, b) a numeric up down to input the number of attendees, c) a numeric up down to input the number of nights, d) radio buttons to indicate the accommodation choice - Single, Double or Suite, and d) check boxes for optional services – Internet Access **and/or** Recreation Center Access (Note: the user can choose one or both or neither of the optional services).

Write code or set properties to ensure the following:

The user inputs a conference name in the text box
The user selects one of the accommodation choices
The minimum number of attendees is 15 and the maximum is 850
The minimum number of nights is 3 and the maximum is 30

The Click event for a Create Conference button should

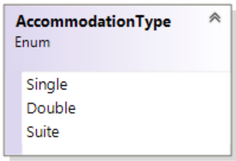
- Get the values from the input controls and assign them to variables
- Instantiate a Conference object
- Format and display the accommodation charge, optional services charge, discount amount, and total charge in one or more labels.
- Disable the Create Conference button and conference name text box
- Enable a Modify Conference Event button (this button should be initially disabled)

The Click event for the Modify Conference button should

- Reset the properties of the conference object (the object that was instantiated in the Create Conference button click method – see above). **Note:** you do not have to check which inputs (i.e., accommodation choice, number of attendees, number of nights and optional services), if any, were changed prior to the user clicking the Modify Conference button. Assume the user had changed all inputs (except for conference name).
- Format and display the revised accommodation charge, optional services charge, discount amount, and total charge.

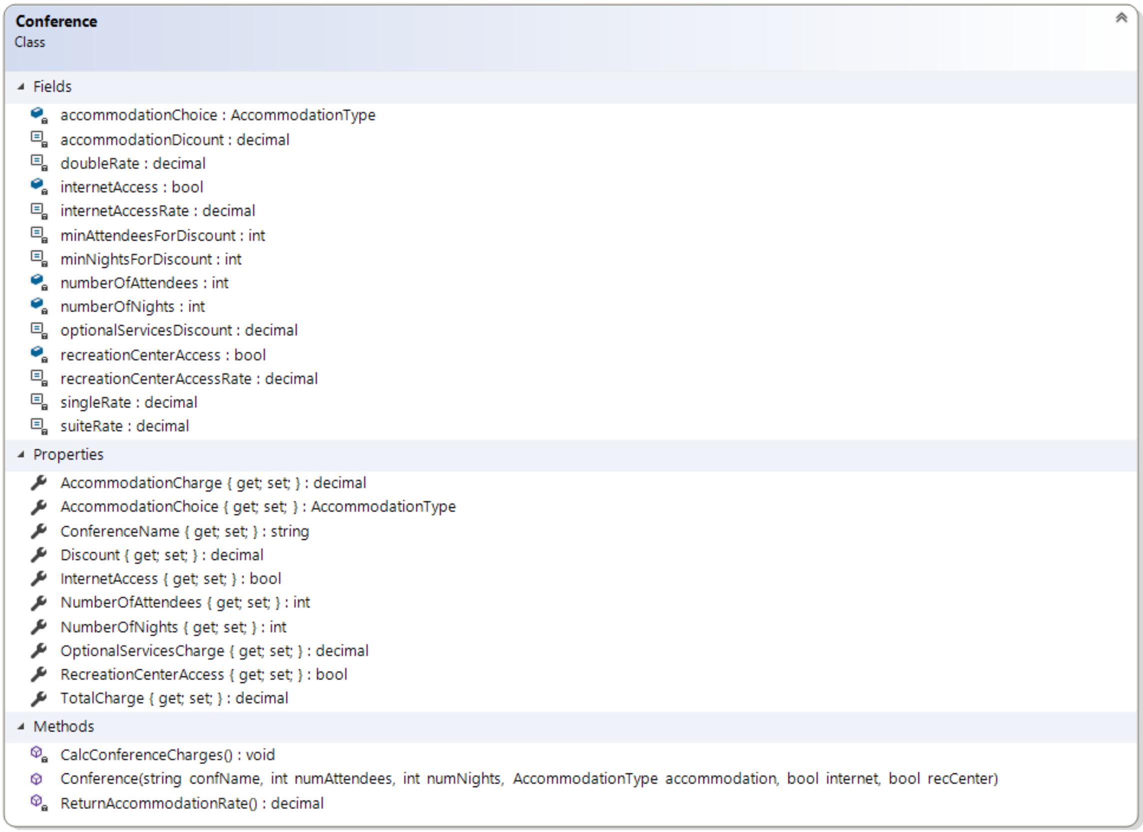
Include a Clear button to clear or reset the input and output values, enable the Create Conference button, and disable the Modify Conference button.

Include an Exit button to exit the application. Use a message box to confirm that the user really wishes to exit the application before closing the form.



AccommodationType
Enum

Single
Double
Suite



Conference
Class

Fields

accommodationChoice : AccommodationType
accommodationDicount : decimal
doubleRate : decimal
internetAccess : bool
internetAccessRate : decimal
minAttendeesForDiscount : int
minNightsForDiscount : int
numberOfAttendees : int
numberOfNights : int
optionalServicesDiscount : decimal
recreationCenterAccess : bool
recreationCenterAccessRate : decimal
singleRate : decimal
suiteRate : decimal

Properties

AccommodationCharge { get; set; } : decimal
AccommodationChoice { get; set; } : AccommodationType
ConferenceName { get; set; } : string
Discount { get; set; } : decimal
InternetAccess { get; set; } : bool
NumberOfAttendees { get; set; } : int
NumberOfNights { get; set; } : int
OptionalServicesCharge { get; set; } : decimal
RecreationCenterAccess { get; set; } : bool
TotalCharge { get; set; } : decimal

Methods

CalcConferenceCharges() : void
Conference(string confName, int numAttendees, int numNights, AccommodationType accommodation, bool internet, bool recCenter)
ReturnAccommodationRate() : decimal

The fields with this icon ▤ are constants.

Program 10

Create a project to calculate property tax based on a) square footage of the building (assume one building per property), b) square footage of the land, c) age of the building, and d) location of the property.

The form class should have appropriate controls for entering the following data:

Data	**Required Validation**
Property owner name	cannot be blank 30 characters or less in length
Building square footage	positive integral value
Land square footage	positive integral value
Year built	positive integral value <= current year
Location	One of the following: city, suburb or rural

Building tax is calculated per the rate table shown below:

Building size (square feet)	**Rate**
1-1000	$1.11/square foot
1001-2000	$1110 + $1.13 for each square foot greater than 1000
2001-3000	$2240 + $1.15 for each square foot greater than 2000
3001-4000	$3390 + $1.17 for each square foot greater than 3000
>4000	$4560 + $1.19 for each square foot greater than 4000

Land tax is calculated per the rate table shown below:

Land size (square feet)	**Rate**
1-10000	$0.12/square foot
10001-20000	$1200 + $0.14 for each square foot greater than 10000
20001-30000	$2600 + $0.16 for each square foot greater than 20000
30001-40000	$4200 + $0.18 for each square foot greater than 30000
>40000	$6000 + $0.20 for each square foot greater than 40000

If the building is more than 15 years old, the following deduction is applicable:

- 0.40 percent of the building tax is deducted for each year of the building's age
- For example, if current year = 2024, year built = 2000 and building tax = $5000; then deduction = 24 * (0.004 * 5000) = $480

If the property's location is rural, the following deduction is applicable:

- 3.75 percent of the land tax is deducted for land sizes less than or equal to 28,000 square feet
- 2.15 percent of the land tax is deducted for land sizes greater than 28,000 square feet

PropertyTax Class

- 1 Enumeration for Location Type – City, Suburb, Rural (**note:** place this enumeration in the PropertyTax.cs file, but outside the class block

- 5 Instance Properties (read-only fields)
  - Name of the property owner
  - Building square footage
  - Land square footage
  - Year built
  - Location of property

- 6 Instance Properties (auto-implemented)
  - Building age (public get and private set)
  - Building tax (public get and private set)
  - Land tax (public get and private set)
  - Building tax deduction – for buildings older than 15 years (public get and private set)
  - Land tax deduction – for rural properties (public get and private set)
  - Total property tax (public get and private set)

- 1 Constructor
  - To instantiate object and set the five instance properties (read-only fields)
  - Calculate and set the building age property (building age = current year – year built)
  - Call the instance method that calculates and sets the total property tax

- 4 Private Instance Methods
  - To calculate and set the building tax
  - To calculate and set the land tax
  - To calculate and set the building tax deduction
  - To calculate and set the land tax deduction

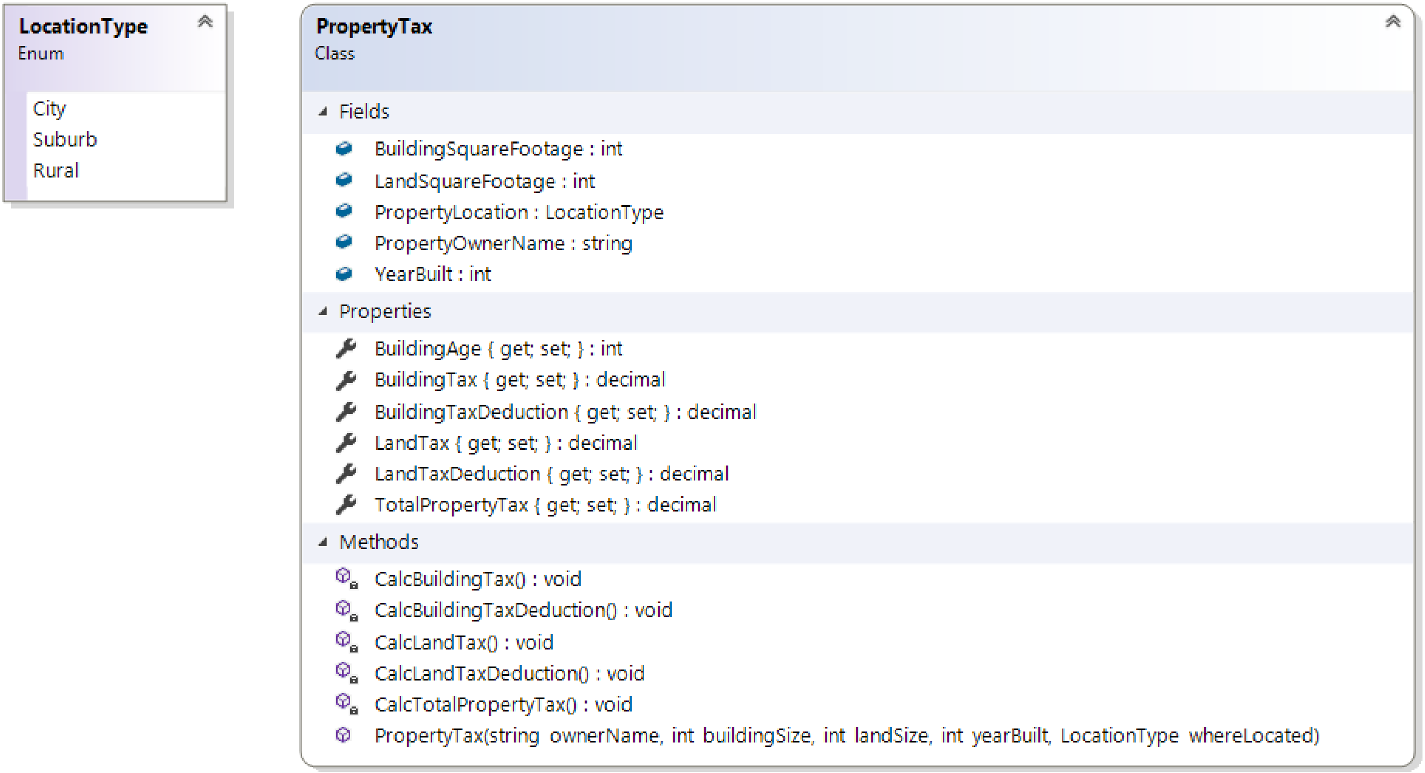
- 1 Private Instance Method
  - To call the private methods that calculate and set the building and land taxes
  - To conditionally (i.e., if the building age > 15) call the method that calculates and sets the building tax deduction
  - To conditionally (i.e., if the property's location is rural) call the method that calculates and sets the land tax deduction
  - To calculate and set the total property tax (building tax + land tax – building tax deduction – land tax deduction).

The Click event for a Create Property Tax button (in the Form class) should

- Validate the input data
- Instantiate a PropertyTax object
- Format and display in one or more labels the building age, building tax, land tax, building tax deduction, land tax deduction, and total property tax.

Include a Clear button to clear or reset the input and output values

Include an Exit button to exit the application.



LocationType
Enum

City
Suburb
Rural

PropertyTax
Class

Fields

BuildingSquareFootage : int
LandSquareFootage : int
PropertyLocation : LocationType
PropertyOwnerName : string
YearBuilt : int

Properties

BuildingAge { get; set; } : int
BuildingTax { get; set; } : decimal
BuildingTaxDeduction { get; set; } : decimal
LandTax { get; set; } : decimal
LandTaxDeduction { get; set; } : decimal
TotalPropertyTax { get; set; } : decimal

Methods

CalcBuildingTax() : void
CalcBuildingTaxDeduction() : void
CalcLandTax() : void
CalcLandTaxDeduction() : void
CalcTotalPropertyTax() : void
PropertyTax(string ownerName, int buildingSize, int landSize, int yearBuilt, LocationType whereLocated)