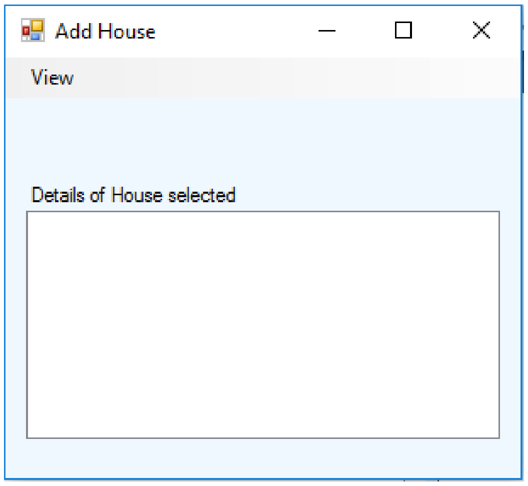
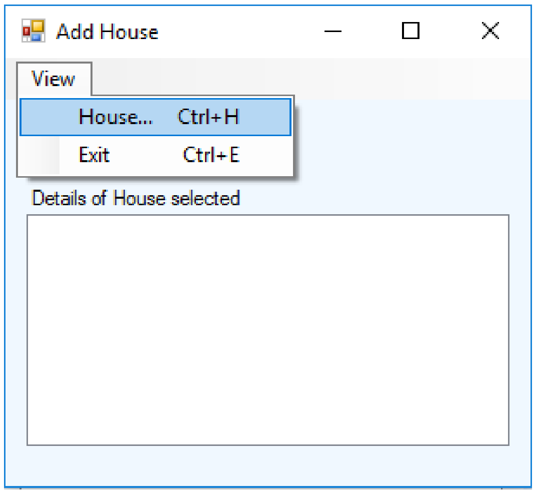
Project #1 (Check Blackboard for due date) 8% of Final Mark

Description: Create the application that shows the houses on sale. You are also required to calculate the total and average price for all the houses.

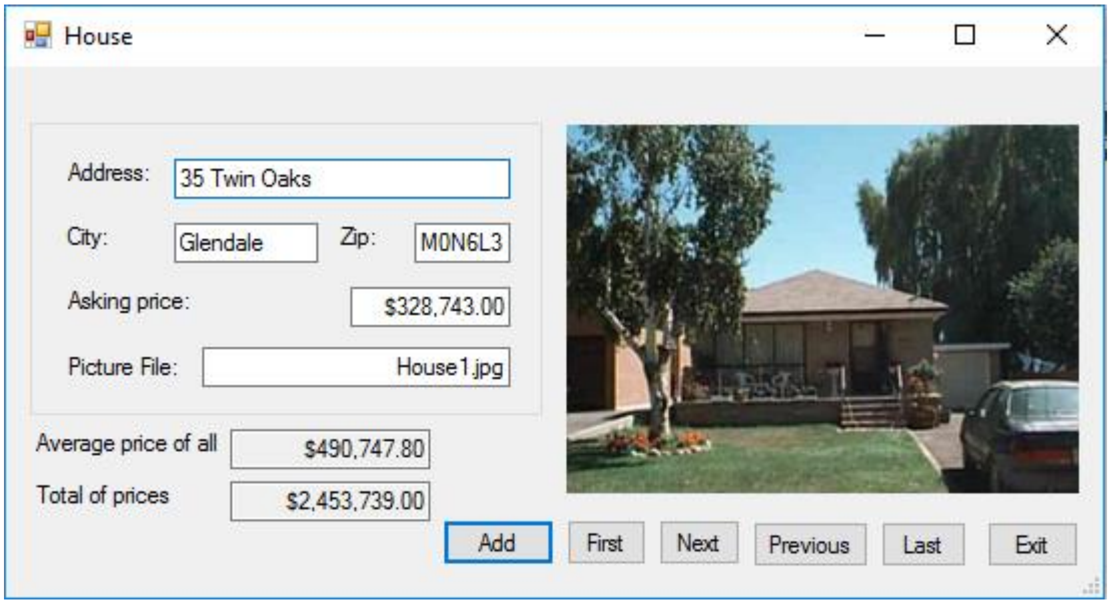
Project starts with 'frmAddHouse' (Add House) as displayed:



User will view the house on sale (short cut keys should work as well):



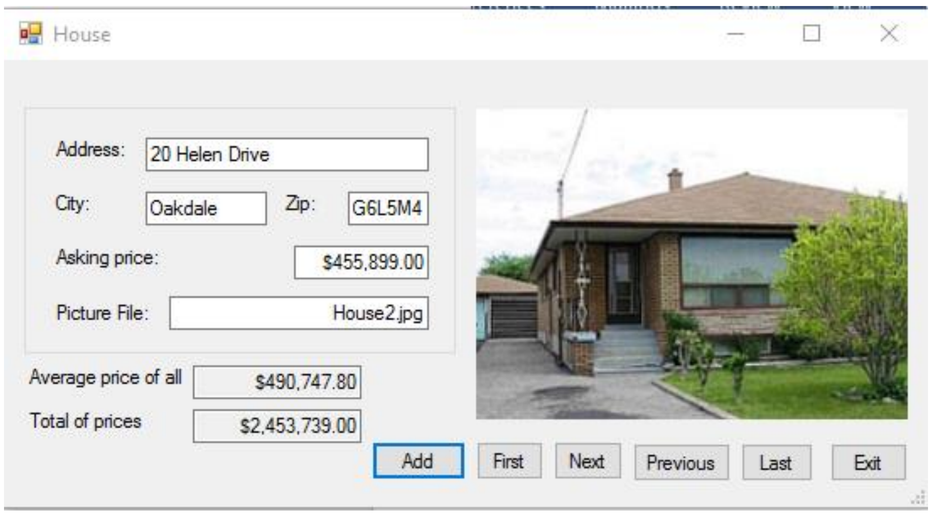
When user chooses 'House', second form loads. The first house on sale is shown at load time along with the average and total price of all:



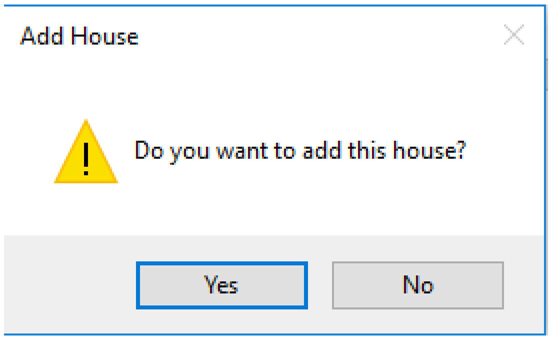
Address: 35 Twin Oaks
City: Glendale
Zip: M0N6L3
Asking price: $328,743.00
Picture File: House1.jpg
Average price of all: $490,747.80
Total of prices: $2,453,739.00

User can view all the houses on sale by keep clicking *Next* button, can move to previous house by hitting *Previous* button, see the last and first on list as he/she clicks *Last* and *First* button

User chooses a house of interest and clicks *Add* button:


House

Address: 20 Helen Drive
City: Oakdale Zip: G6L5M4
Asking price: $455,899.00
Picture File: House2.jpg
Average price of all $490,747.80
Total of prices $2,453,739.00


Add House

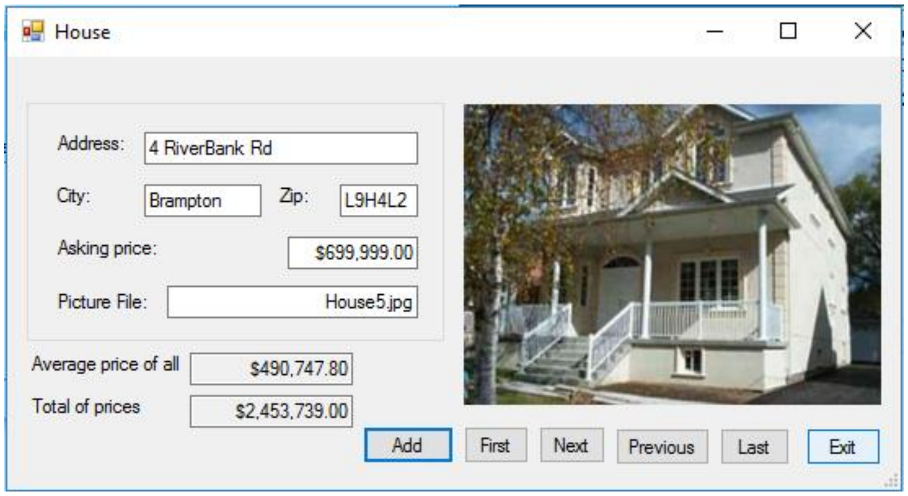
Do you want to add this house?

When user clicks 'Add', message box appears with confirmation:

User clicks 'Yes', data is stored and form resets to next house on sale

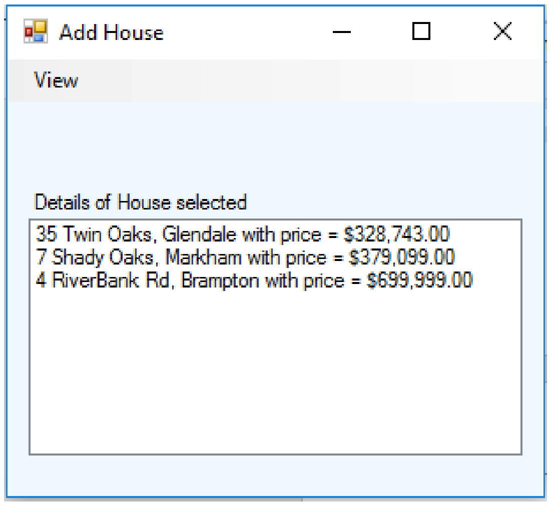
User chooses another house and clicks Add.

User is finished, 'exits' this form to see selected properties (short cut key should work as well):


House

Address: 4 RiverBank Rd
City: Brampton Zip: L9H4L2
Asking price: $699,999.00
Picture File: House5.jpg
Average price of all $490,747.80
Total of prices $2,453,739.00

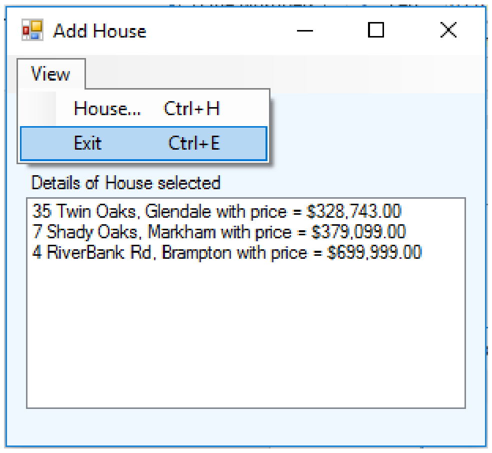
User is then returned to ‘frmAddHouse’ , with address and price:



Details of House selected

```
35 Twin Oaks, Glendale with price = $328,743.00
7 Shady Oaks, Markham with price = $379,099.00
4 RiverBank Rd, Brampton with price = $699,999.00
```

User wishes to ‘exit’ the program:



Details of House selected

```
35 Twin Oaks, Glendale with price = $328,743.00
7 Shady Oaks, Markham with price = $379,099.00
4 RiverBank Rd, Brampton with price = $699,999.00
```

Additional Requirements:

Form Design:

- Marks will be allocated for:
  - Accelerators
  - Exit menu items have separator above them
  - Shortcuts on menu items
  - Accept and cancel buttons
  - Correct message box capitalization
  - Tab stops set correctly
  - Correct controls on forms
  - Spelling and grammar

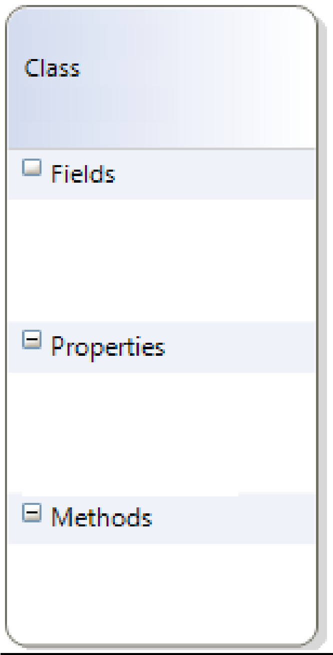
Overall Programming Practices :

- Use conventional naming practices for your variables, fields, properties, and methods.
- Comment your code with a brief description of what is going on.
- Re-use code wisely. If you find you are using the same code in more than one place, create a method and call it

- Hard Coding: Do NOT store data or hard code branch information in your forms! You may hardcode values in your classes. The hard coded data in your class will act as our database to grab the house details. When you need house details in your forms, use programming practices / methods to get them. The same principle applies with 'average and totals' – store the data in your classes and access it from your forms using programming practices / methods.
- **There should be two classes in class library (House and HouseList).** Decide what fields and properties describe a house and put them in house class. You can decide whether to use automatic or standard properties.
- HouseList class should hold methods to retrieve details about the house. It should have a list to hold house data and some service methods like:
  - A method to display the first house when forms loads.
  - Calculate and return the average and total price of houses.
  - For houses which are not yet added to the listbox, takes them and returns them in a list of strings. Hint: take a Boolean status for house and change it if its added to the listbox.
  - These are few suggestions but you need few more methods in the HouseList class.
- Remember most of the processing should be in the classes and call it in the form to access class data.

UML Diagram Format:

- Remember that a C# UML diagram looks like this:


Class
Fields
Properties
Methods