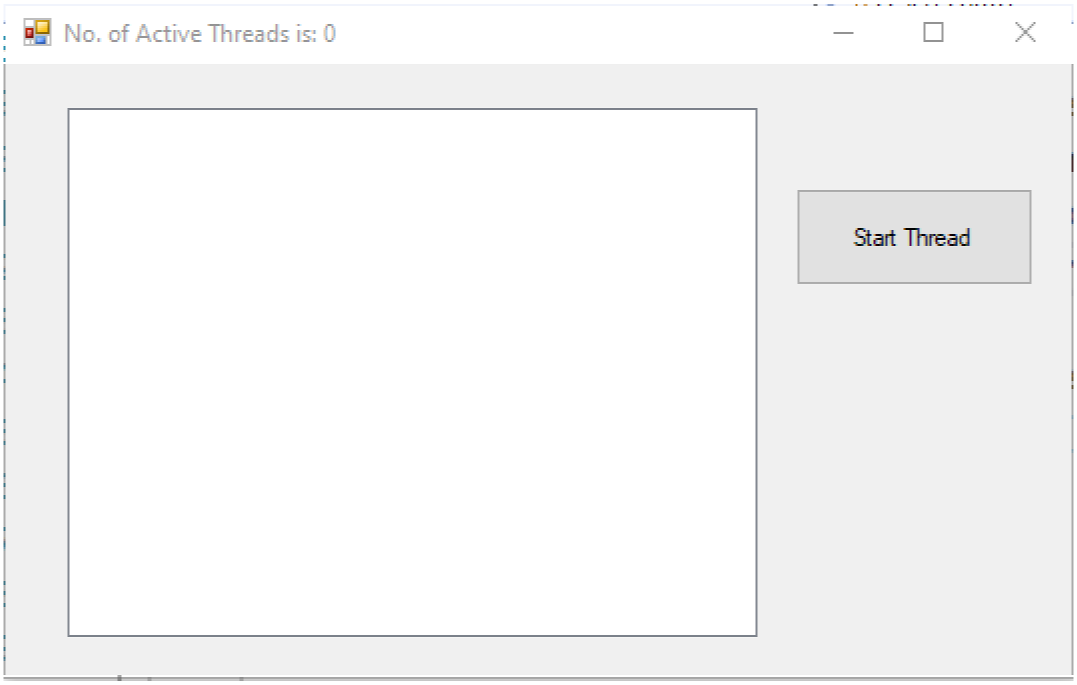
CMPE 1666- Practice Questions for Lab Exam3 – Fall 2024

Question 1-

Create a form-based application with the controls shown below. As the form grows, the list box should grow with it both in width and height. The Form also contains a timer, with an interval of 100ms.



The Form class must contain a stopwatch object and declare a list of Threads.

The Form load event will create a GDIDrawer window, create the empty list of Threads (list that was declared in the Form class) and start the stopwatch. **Note that the global handle to the GDI Drawer can be used in multiple threads directly.**

Each time the user clicks on the button "**Start Thread**", a new thread will be created and set as background thread. The thread will create a ball (represented by a circle) at a random y position on the left end of the GDIDrawer window. The thread will sleep for 100 ms, wake up and move the ball by a random number of pixels, between 10 and 40, towards the right (see video). The ball will move until it reaches the right end of the GDIDrawer window, after which it disappears. At each movement of the ball, the thread will update the list box with its name (i.e the Thread's name), the time elapsed in milliseconds (since the application started – using the **stopwatch**) and the (x,y) position of the center of the ball. Use a delegate for this purpose. Give the threads names that are easy to generate eg. Thread1, Thread2, Thread3 etc. At any one time, we expect to have several threads displaying balls.

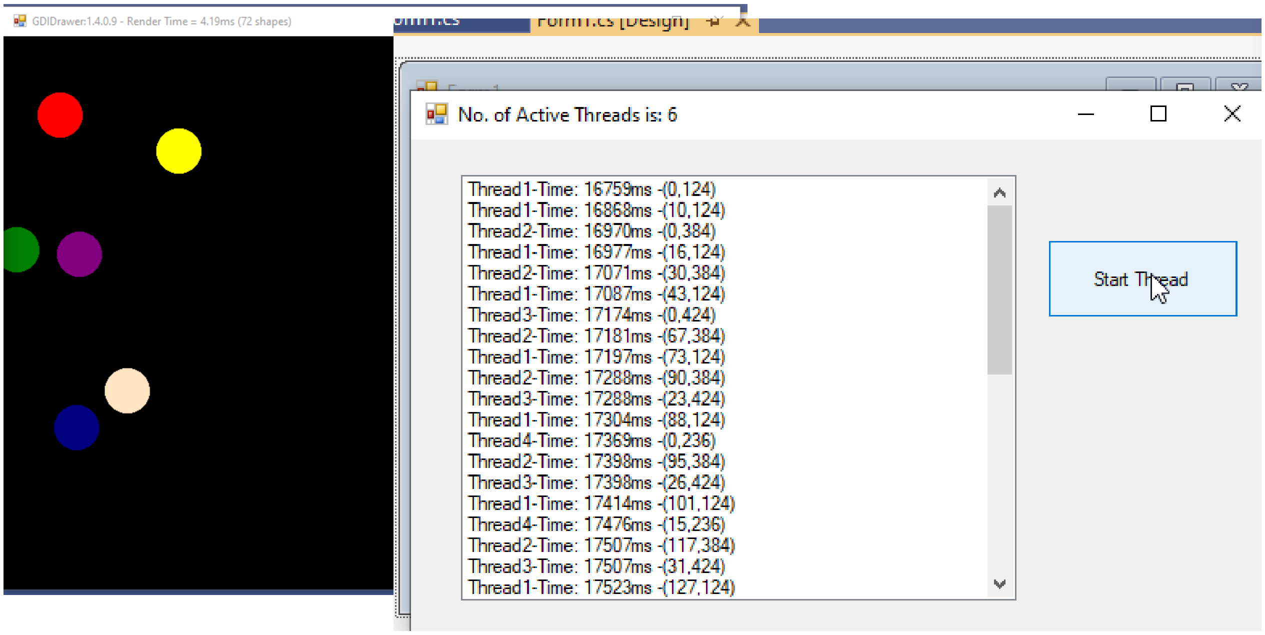
You can use the **AddCenteredEllipse()** method or any other suitable method from **GDIDrawer** to draw the balls. You can erase a ball from its previous position by covering it with a black ball of the same size.

Each thread, when created, must be added to the thread list (declared in the Form class and created at the Form-load event). When a thread has stopped, it must be removed from the list. At each timer tick the Form text should display the number of currently active threads (You must again use a delegate here).

The colors of the balls must be random. You can choose to keep a few colors (say 10) in an array and choose randomly between these colors.

Note: Both the System.Diagnostics class and the System.Threading class have a **ThreadState** class. To avoid ambiguity, ensure that you use **System.Threading.ThreadState,** when you need to use the ThreadState class.

A snapshot of Sample Run

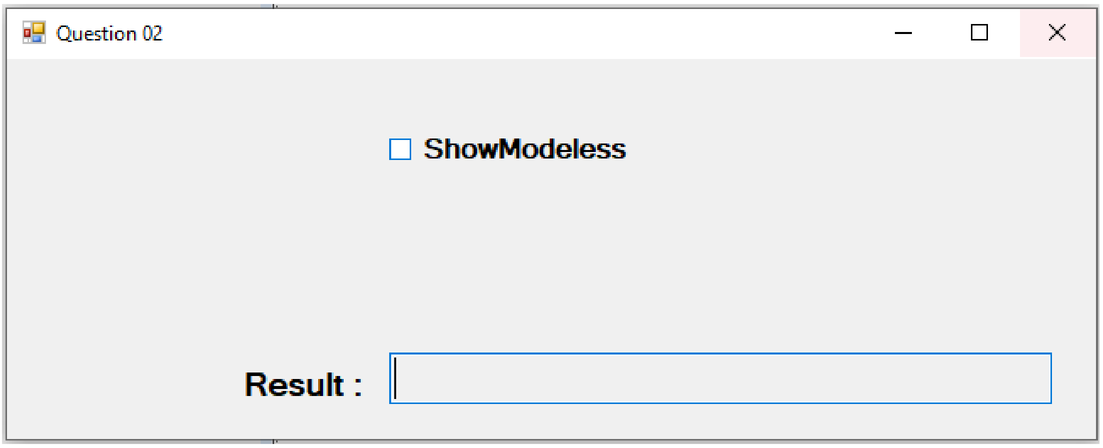

GDIDrawer:1.4.0.9 - Render Time = 4.19ms (72 shapes)
Form1.cs [Design]
No. of Active Threads is: 6
Start Thread

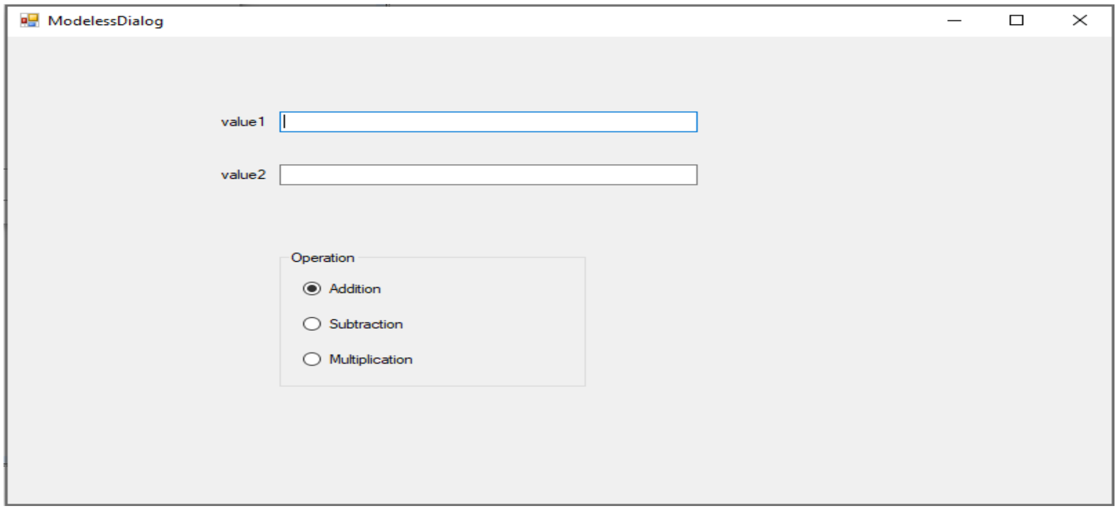
```
Thread1-Time: 16759ms -(0,124)
Thread1-Time: 16868ms -(10,124)
Thread2-Time: 16970ms -(0,384)
Thread1-Time: 16977ms -(16,124)
Thread2-Time: 17071ms -(30,384)
Thread1-Time: 17087ms -(43,124)
Thread3-Time: 17174ms -(0,424)
Thread2-Time: 17181ms -(67,384)
Thread1-Time: 17197ms -(73,124)
Thread2-Time: 17288ms -(90,384)
Thread3-Time: 17288ms -(23,424)
Thread1-Time: 17304ms -(88,124)
Thread4-Time: 17369ms -(0,236)
Thread2-Time: 17398ms -(95,384)
Thread3-Time: 17398ms -(26,424)
Thread1-Time: 17414ms -(101,124)
Thread4-Time: 17476ms -(15,236)
Thread2-Time: 17507ms -(117,384)
Thread3-Time: 17507ms -(31,424)
Thread1-Time: 17523ms -(127,124)
```

Question 2- Modeless Dialogs

Delegates must be used where appropriate for this question

1. Create A Form-based application that has a main form and a Modeless Dialog as shown in the figure. The main form consists of a checkbox and a read-only textbox. The modeless dialog consists of 2 textboxes and a group box containing 3 radio buttons. Each of the form controls contains a corresponding label.
2. The 2 textboxes on the modeless dialog are for input of values from the user and the read-only textbox on the main form is for the display of results. The operation performed on the values will be either addition, subtraction (value1-value2) or multiplication, based on the radio button that is checked.
3. The **Addition** radio button must be selected by default





Operation

● Addition

○ Subtraction

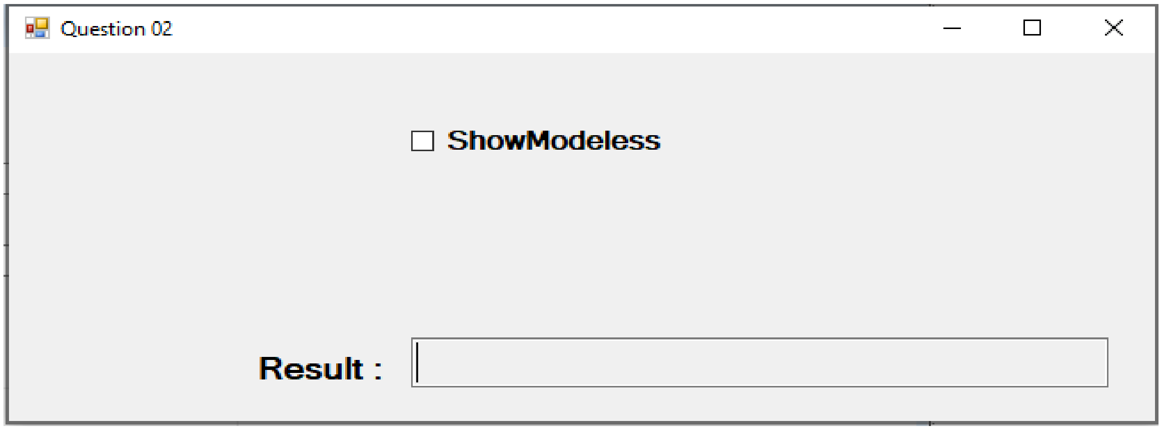
○ Multiplication

4. The “**Show Modeless**” checkbox (on the main form), when checked must cause the Modeless dialog to appear.
5. If the checkbox is unchecked it must cause the Modeless dialog to hide, but not destroyed.
6. Using delegates, when the user inputs values for the text boxes in the Modeless Dialog box, an operation will be performed, whenever **both** textboxes have valid values. The operation performed will be based on the radio button that is checked. The result of the operation must appear in the read-only result textbox on the main form. Any change in values in the Modeless Dialog as well as any click on the radio buttons must immediately be reflected in the result value displayed on the main form. **Use event consolidation**.
7. At any time if any of the input textboxes is empty or contains an invalid value, the result in the result textbox must be 0.
8. When the user clicks on the control X in the modeless dialog form, the form must be hidden (instead of closing) and the checkbox on the main form must be automatically unchecked.

9. The user should be able to open and close the Modeless dialog any number of times. Note: Because the modeless dialog is hidden, not closed, any values chosen before hiding the dialog will appear when the form is displayed again.

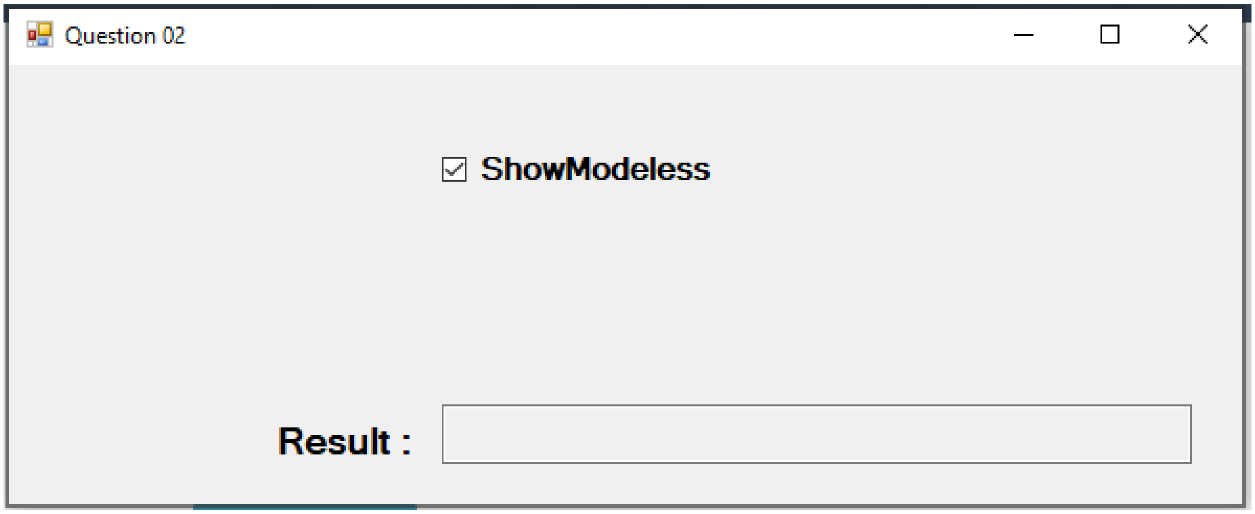
Sample Runs

Program starts running- Checkbox unchecked- Result Empty

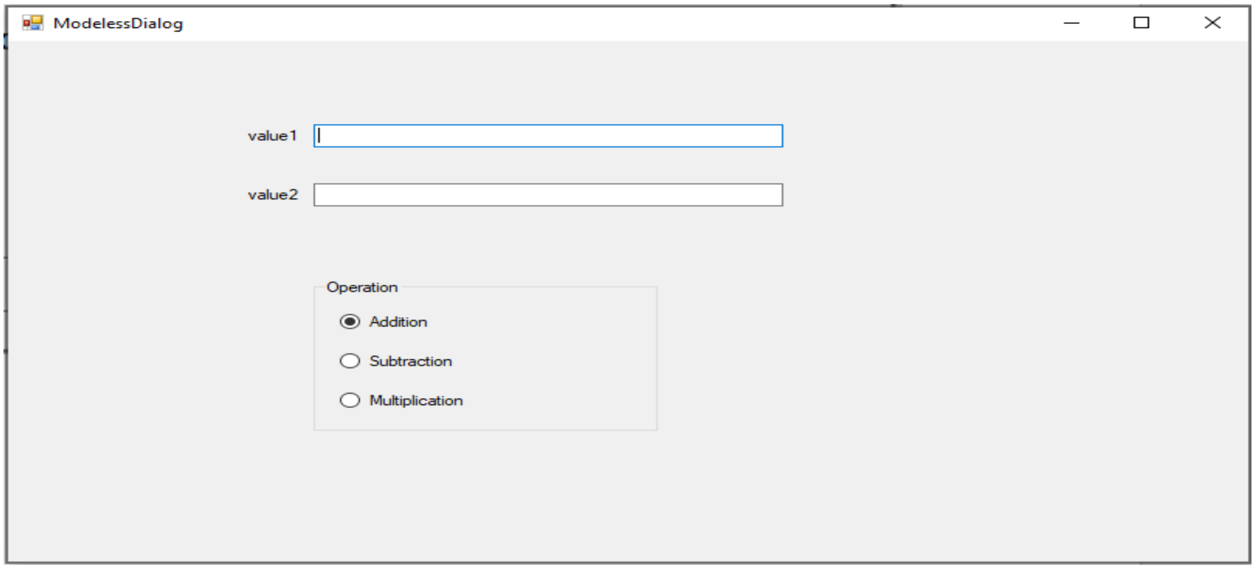


□ ShowModeless

Check on checkbox causes modeless dialog to appear- Note: Addition is selected by default



☑ ShowModeless



ModelessDialog

value1

value2

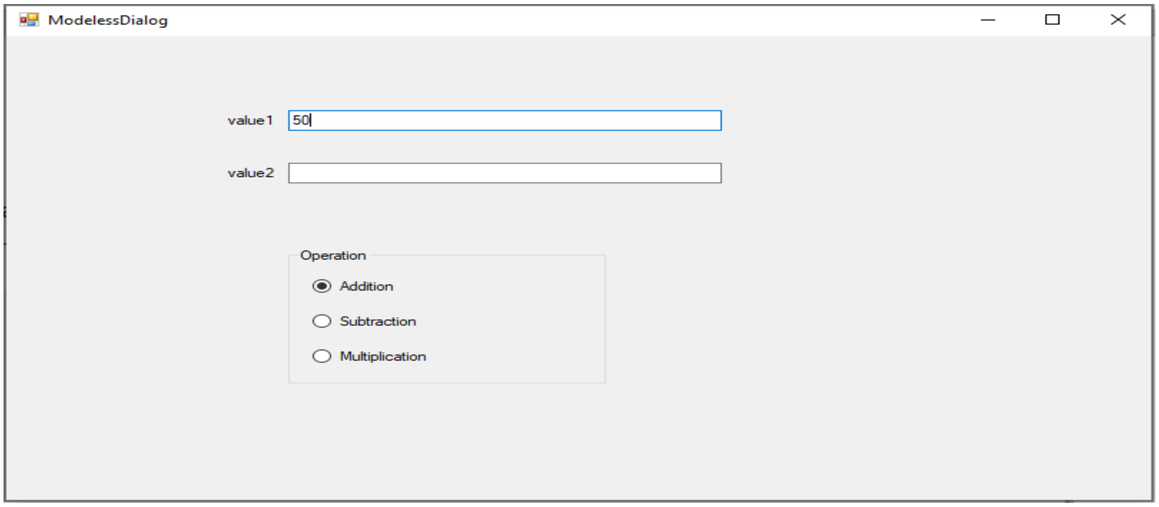
Operation

● Addition

○ Subtraction

○ Multiplication

First textbox contains a valid value, second one is empty- result is 0.



ModelessDialog

value1 50

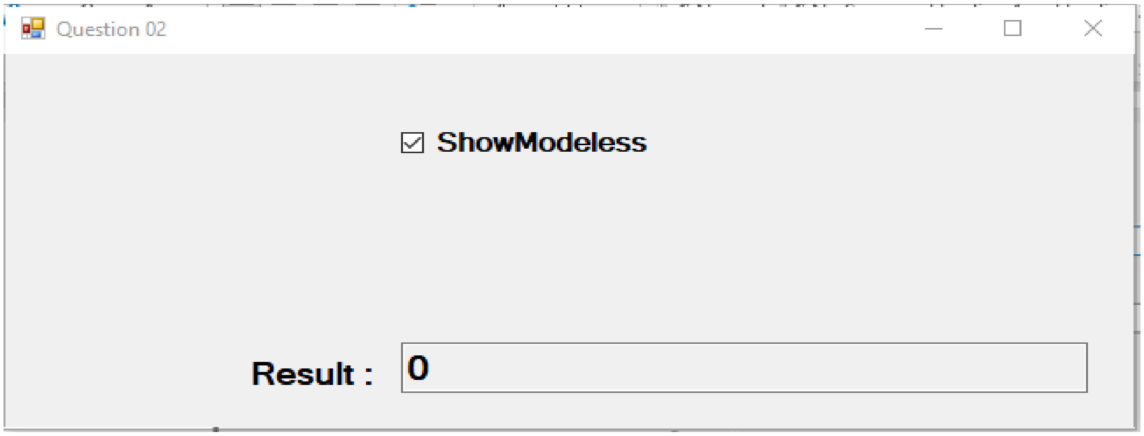
value2

Operation

● Addition

○ Subtraction

○ Multiplication

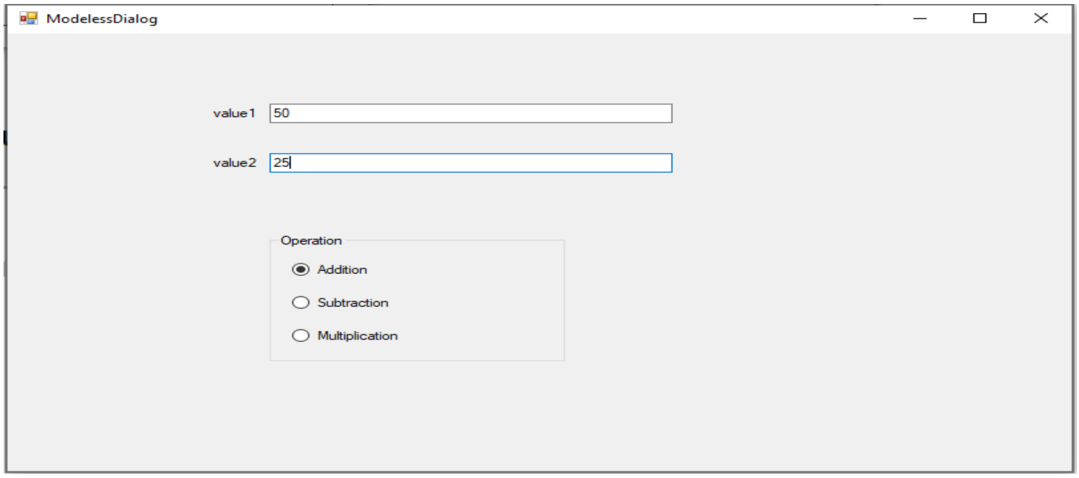


Question 02

☑ ShowModeless

Result : 0

Both textboxes contain valid values, Result box contains result of operation based on the selected radio button



value1 50

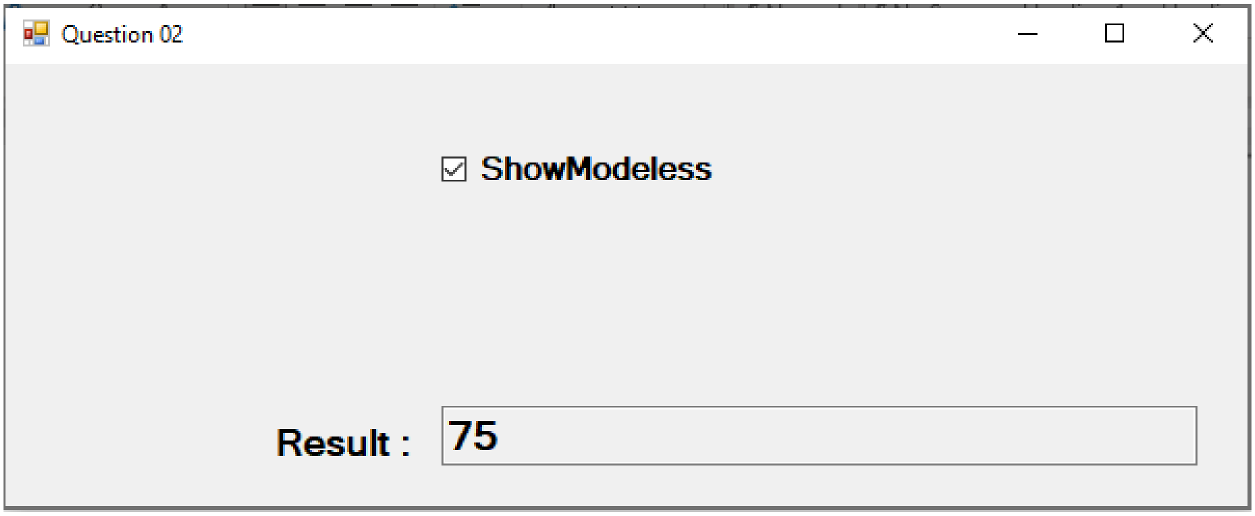
value2 25

Operation

● Addition

○ Subtraction

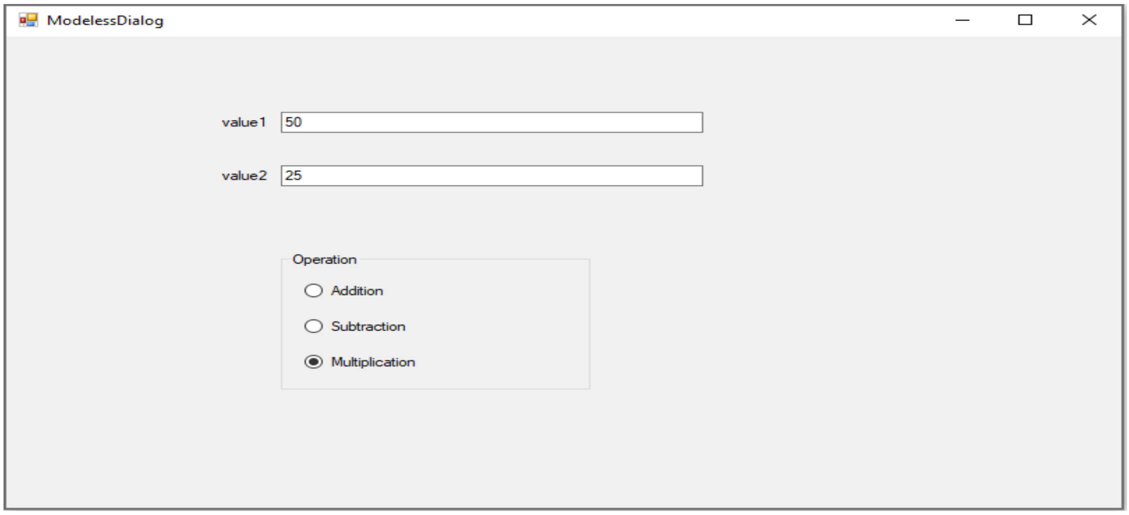
○ Multiplication



☑ ShowModeless

Result : 75

Change in Selected Operation causes immediate change in result



ModelessDialog

value1 50

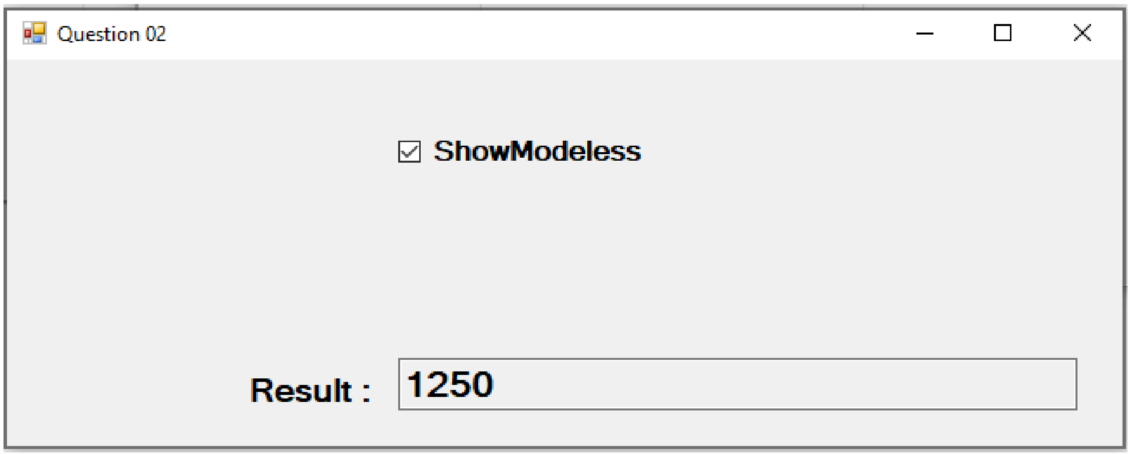
value2 25

Operation

○ Addition

○ Subtraction

● Multiplication



Question 02

☑ ShowModeless

Result : 1250