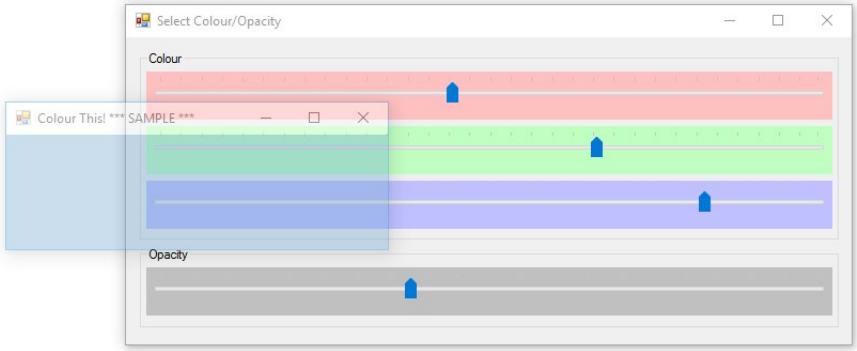
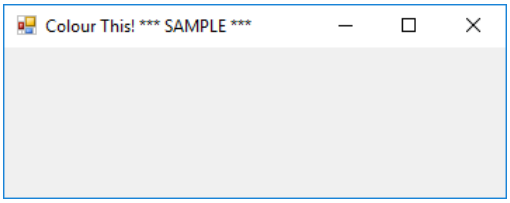
School of Applied Sciences And Technology
Department of IST
Program: CNT

CMPE1666- ICA 15 – Color This!

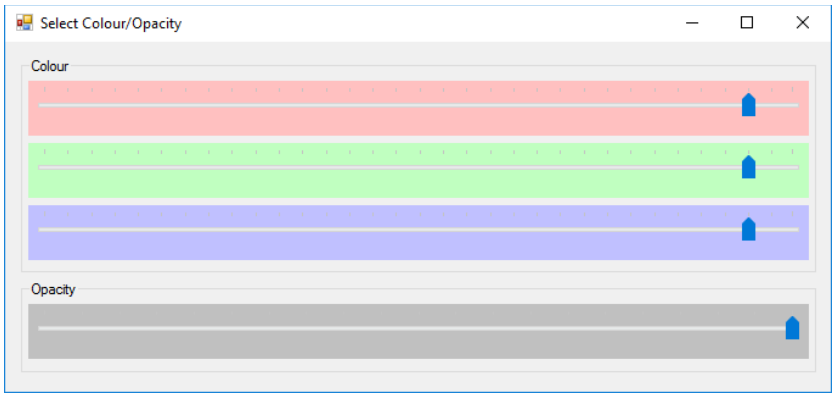


In this ICA you will exercise your understanding of modeless dialogs, delegates, and public methods.

You will build a simple forms application that has no controls on it:



When the user clicks anywhere in the form, a modeless dialog with the following appearance will be show:



The three upper track bars will control the RGB intensity for the background color of the main form. The values for each trackbar control (each RGB colour component) must reflect the current background color when the dialog is shown. Changing any of the track bars will immediately be reflected as a change in the background color of the main form.

The bottom track bar will likewise control the opacity of the main form. Note that opacity is set in a form as a **`double`**, and represents opaque range as a fraction 0.0 to 1.0. Your opacity track bar will store an **`int`**, from 0-100. You will need to do some converting to make this work correctly.

NOTE: This control represents the opacity of the main form, NOT the alpha channel for the background colour of the main form, as you might expect.
Closing the modeless dialog via the "X" will only hide the dialog. Clicking on the main form will bring the dialog back.

Notes

As with the notes/demos, you will need to add delegates, public methods and callbacks in this application. Ensure that you understand what needs to be done before you begin.

While the **`Color`** type supports an alpha channel, the BackColor property of the main form does not. You should pass the RGB components to the modeless dialog as a color, and pass the opacity separately. This way you will only need two public methods in the modeless dialog. You can use these methods to establish initial positions for the track bars.

It would be simplest to create a single callback for any RGB change. If any of the track bars change position, build a full RGB color (using `Color.FromArgb`) and invoke the callback with a full RGB color. This will be easier than creating three callback that take individual R, G, and B values. Event consolidation would then work nicely, too.

You will need callbacks in the main form for handling the track bar changes. The callbacks should be registered in delegates in the dialog (subscriber delegates), immediately after creation of the dialog.

Main Form:

- Create the modeless dialog
- Create callback functions to handle track bar changes
- Register callbacks in delegates in the modeless dialog
- When clicked, use public methods to set values/control properties in the modeless dialog then show it

Modeless Dialog:

- Declare delegate type
- Create delegate members for required actions
- Add public methods to set track bar positions
- When a track bar changes invoke the appropriate delegate, if subscribed

Build this application in incremental steps; perhaps get a single track bar working before you attempt doing the rest.

Rubric – Max Marks: 30

This application will require visual inspection of functionality and code.
Mark loss is at your instructor's discretion but will be applied consistently across all students.

Item	Marks	Penalties
UI Design (40%) • UI is as directed. • Tab Order is Correct. • Effective use of space in dialog • Initial form colour presented in dialog. • Colour changes in main form are immediate. • Dialog hide/show behavior is correct.	2 1 1 2 4 2	
Code Design and Implementation (60%) • Appropriate consolidation for events and support methods. • All button functions are as described. • Appropriate delegate declaration and usage. • All operations on all events work as expected.	4 5 5 4	
Documentation: • Programmer Block • Well commented code • Appropriate Variable Names • Proper spacing between blocks of code • Control names are consistent and appropriate.		Missing components of documentation: -1 to -6