HW01 Problem Set

CS-3020

Assignment
Deitel and Deitel Exercises 2.9, 3.28, 4.14

HW01-1: (Deitel & Deitel Exercise 2.9)

2.9 *(Calculator GUI)* Create the GUI for the calculator as shown in Fig. 2.42.

a) ***Manipulating the Form's properties.*** Change the Text property of the Form to Calculator. Change the Font property to 9pt Segoe UI. Change the Size property of the Form to 258, 210.

b) ***Adding a TextBox to the Form.*** Set the TextBox's Text property in the **Properties** window to 0. Stretch the TextBox and position it as shown in Fig. 2.42. Set the TextAlign property to Right—this right aligns text displayed in the TextBox.

c) ***Adding the first Panel to the Form.*** Panel controls are used to group other controls. Add a Panel to the Form. Change the Panel's BorderStyle property to Fixed3D to make the inside of the Panel appear recessed. Change the Size property to 90, 120. This Panel will contain the calculator's numeric keys.

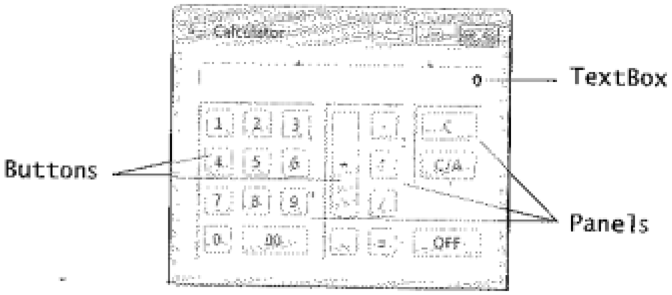
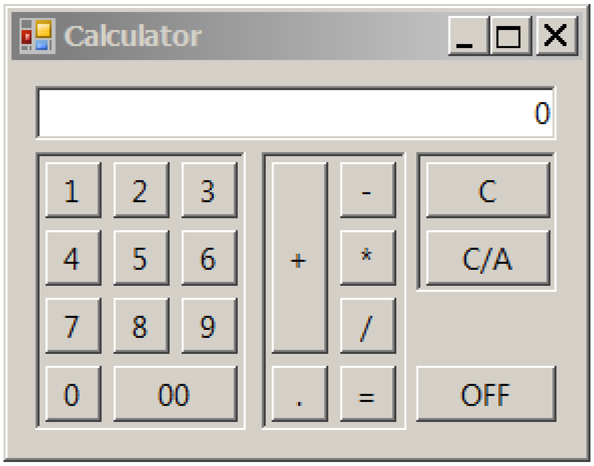


Fig. 2.42 | Calculator GUI.

d) ***Adding the second Panel to the Form.*** Change the Panel's BorderStyle property to Fixed3D. Change the Size property to 62, 120. This Panel will contain the calculator's operator keys.

e) ***Adding the third (and last) Panel to the Form.*** Change the Panel's BorderStyle property to Fixed3D. Change the Size property to 54, 62. This Panel contains the calculator's **C** (clear) and **C/A** (clear all) keys.

f) ***Adding Buttons to the Form.*** There are 20 Buttons on the calculator. Add a Button to the Panel by dragging and dropping it on the Panel. Change the Text property of each Button to the calculator key it represents. The value you enter in the Text property will appear on the face of the Button. Finally, resize the Buttons, using their Size properties. Each Button labeled 0–9, *, /, -, = and . should have a size of 23, 23. The **00** Button has size 52, 23. The **OFF** Button has size 54, 23. The + Button is sized 23, 81. The **C** (clear) and **C/A** (clear all) Buttons are sized 44, 23.



HW01-2: (Deitel & Deitel Exercise 3.28)

3.28 *(Digits of an Integer)* Write an app that inputs one number consisting of five digits from the user, separates the number into its individual digits and displays the digits separated from one another by three spaces each. For example, if the user types in the number `42339`, the app should display

```
4   2   3   3   9
```

Assume that the user enters the correct number of digits. What happens when you execute the app and type a number with more than five digits? What happens when you execute the app and type a number with fewer than five digits? [*Hint:* It's possible to do this exercise with the techniques you learned in this chapter. You'll need to use both division and remainder operations to "pick off" each digit.]

HW01-3: (Deitel & Deitel Exercise 4.14)

4.14 *(`Date` Class)* Create a class called `Date` that includes three pieces of information as automatic properties—a month (type `int`), a day (type `int`) and a year (type `int`). Your class should have a constructor that initializes the three automatic properties and assumes that the values provided are correct. Provide a method `DisplayDate` that displays the month, day and year separated by forward slashes (/). Write a test app named `DateTest` that demonstrates class `Date`'s capabilities.

Supplemental Information

Be sure to review and adhere to the documentation and submission guidelines (see course website).

Hints and Suggestions

Exercise 2.9: The text should be centered nicely in the buttons. Using Visual Studio Express 2013 I found that the font and text size they specified resulted in the button labels being below center and being slightly cut off. A height of 28 seemed to work well. Since most of the buttons are supposed to be square, this means making those 28x28. Don't forget to scale all of the other elements appropriately so that the overall appearance is close to that shown in the text.

Exercise 3.28: While not specified in the assignment, your program should display leading zeroes if the number entered is fewer than five digits and it should display just the last five digits if the number is over five digits. Remember that, at this point, we have not covered control structures (if-else or loops) so the intent is for you to just have a sequence of statements that accomplishes the task. Don't make this one harder than it is.

Exercise 4.14: Although it is implied, it isn't explicitly stated that your constructor should take three integer arguments, namely the month, day, and year. Don't ignore the specification that you are supposed to use automatic variables. Also, consider which variables, properties, and methods should be public and which should be private.

Grading Rubric

Each problem is worth 10 pts (score will be recorded as a percentage of that amount)

10% Properly submitted
10% Properly named
20% Adequate comments
10% Runs
20% Produces correct output
30% Effort evidenced by the submitted work