

Inft2012 Application Programming – Notes for week 6

Pair programming

Remember, programming in pairs – and taking turns at the keyboard – is known to be beneficial when learning to program and to enhance productivity in actual programming. You are strongly encouraged to work with a partner on the exercises each week. The assignment will be set next week: have you decided who to pair up with?

The mark of the web

When you try to run this week's demo program, you will probably be told that you can't – because AboutBox.resx has 'the mark of the web'. Sounds ominous, doesn't it!

The mark of the web is a Visual Studio initiative to stop certain files from running if they come from elsewhere. I'm still not entirely clear which files VS decides to apply this mark to, but it has clearly applied it to AboutBox.resx.

To fix the problem, find the file AboutBox.resx in Windows Explorer (it will be in the folder of the project that contains lots of files along with the bin and obj and Properties folders). Right-click on the file, and select Properties. With any luck, on the General tab, down at the bottom, there will be some warning text and a checkbox labelled Unblock. If you tick that checkbox and choose OK, the mark of the web should disappear from the file, and you should be able to run the project. You'll probably need to restart Visual Studio, or at least to reload this project, after fixing the file.

You won't be able to practice this on files that don't come from the internet, because they don't have the mark of the web, so they don't get that Security information and checkbox at the foot of the General tab.

Lab exercises

Most students will probably not finish all of these exercises in the lab class. Remembering that learning to program takes lots of practice, you are strongly advised to finish them before next week's lab class, bringing to the class any problems you encounter.

1. Reading and understanding program code is an essential step on the way to writing program code. Most of the code-reading exercises you have been given ask you to explain the purpose of a piece of code. This one asks you what output a piece of code will produce. There's no obvious purpose to the code, but you can still see what it will do, by desk-checking it and working out what values each variable has at what points of the code.

Considering the following code, what will be displayed in the message box when BtnQ1 is clicked? Note that we are not asking you what will be in the title of the message box – just what will be on the body of its form.

```
private void BtnQ1_Click(object sender, EventArgs e)
{
    int iOne = 5, iTwo = 8, iThree = 13;
    Mystery(iOne, out iTwo, ref iThree);
    string sResponse = Convert.ToString(iOne) + ", " +
        Convert.ToString(iTwo) + ", " + Convert.ToString(iThree);
    MessageBox.Show(sResponse, "Your answers");
}
```

```
private void Mystery(int i1, out int i2, ref int i3)
{
    i2 = i1 + i3;
    i3 = i1 + i2;
    i1 = i2 + i3;
}
```

Answer all of the following questions within a single project. For each question add a new form to the project, and add a button to the main form that opens the new form, which then does the task for that question.

2. Add an InputBox to the project, as follows. Have the lecture demo program open at the same time as your project, in two different Visual Studio windows. When you've added a new form called InputBox to your project . . .
 - note the size of the form (in its properties) in the lecture demo, and make your form the same size;
 - select all the form's controls in the lecture demo, copy them, and paste them to your form;
 - set the form's AcceptButton property to BtnAccept (from the drop-down list), its CancelButton property to BtnCancel, and its StartPosition property to CenterParent (or leave these properties with their default values and see what effects this has);
 - check that BtnAccept has a DialogResult property of OK and BtnCancel has a DialogResult property of Cancel – these properties would have copied across when you copied and pasted the buttons;
 - finally, go to the InputBox code in the lecture demo program, copy all of the code between the first and last braces (after the namespace), and paste it over the same bits of code in your InputBox code (ie replace everything between the first and last braces of your code).

You now have a working InputBox. In your program, write some simple code to test it.

3. Write a method that, given a number of seconds (eg 5049) uses integer division and remainder to return the number of hours, minutes, and seconds that it corresponds to. Think about what sort of parameters would best suit such a method. Having called the method, your program (*not* the method) should display the result in a message box in the form "Hours: 1; minutes: 24; seconds: 9".
4. Write a method that, given a number of seconds (eg 5049) returns a single string indicating the number of hours, minutes, and seconds that it corresponds to, in the form "1 hour, 24 minutes, and 9 seconds". Having called the method, your program should display the result in a message box.
5. Write a function method called treble, in three overloaded versions. One version takes a string as a parameter and returns a string consisting of three occurrences of the original string separated by spaces. A second version takes an integer parameter and returns three times that integer. A third version takes a double parameter and returns three times that double.
6. Write a method that takes a double as a parameter and multiplies the parameter by three in such a way that its corresponding argument is also multiplied by 3.