THE UNIVERSITY OF NEWCASTLE AUSTRALIA

School of Information and Physical Sciences

Inft2012 Application Programming – Notes for week 4

Lab exercises

1. Reading and understanding program code is an essential step on the way to writing program code. Read the following code and explain what it does. Try to explain what the purpose of the code might be, rather than its individual steps. Do this just by reading the code – not by entering it into a program and trying it.

   If you can't tell just by reading the code, use pen and paper to work out what it would do with some sample input. The IndexOf lines suggest that the input will have a comma and a space, so see if you can work out what it will do with, say, "Budgerigar, Jonathan Livingston". But once you've done that, don't forget to keep going and explain the overall purpose of the code.

   Note: many students skip these read-understand-explain questions, either because they look too hard or because the other questions look like more fun. There will be read-understand-explain questions on your exam, so you might as well get some practice in!

```
private void BtnTransform_Click(object sender, EventArgs e)
{
    int iPosition;
    string sWorking, sPart2, sResult1, sResult2, sResult3;
    sWorking = TbxIn.Text;
    iPosition = sWorking.IndexOf(",");
    sResult3 = sWorking.Substring(0, iPosition);
    sPart2 = sWorking.Substring(iPosition + 2);
    sResult1 = sPart2.Substring(0, 1);
    iPosition = sPart2.IndexOf(" ");
    sResult2 = sPart2.Substring(iPosition + 1, 1);
    TbxOut.Text = sResult1 + ". " + sResult2 + ". " + sResult3;
}
```

2. In what circumstances will the following boolean expression evaluate to true?

```
sAlphabet.IndexOf(sCharacter) >= 0 &&
         sMessage.IndexOf(sCharacter) < 0
```

   Once you have answered that question, you should be able to complete this description: the expression indicates whether sCharacter is . . . but is not . . .

3. The version of Lec4DemoFormatting on Blackboard has no exception handling, and crashes on various inappropriate date inputs. In the lecture we worked through a gradual improvement to the exception handling. Now work through the lecture again yourself, making the same changes to the program, until it handles any of the date exceptions that were mentioned.

4. Go back to your die-drawing program from last week and add a button labelled “Roll”. When the Roll button is pressed, your program should use the Random class to generate an appropriate number, and then display the corresponding die.

5. The Max function gives the maximum of two numbers. Write a program that finds the maximum of four numbers, each entered in its own textbox. To exercise your mind, don’t use an *if* statement for this exercise.

   Be sure that your program handles exceptions sensibly, so that it doesn’t crash if any of the textboxes doesn’t contain a number.

   Remember, the problem-solving phase is often the hardest part of programming. Don’t even think about writing program code until you’ve worked out how you’re going to use Max to give you the maximum of four numbers.

6. Here is a reasonably typical URL: www.newcastle.edu.au/colleges/schools/sips/staff.html (If you don’t know why it’s called a URL, maybe now would be a good time to find out.) Let’s make up some names for the different parts. Up to and excluding the first slash we’ll call the ‘server’; between and excluding the first and last slashes we’ll call the ‘path’; and after the last slash we’ll call the ‘file’.

   Write a program that accepts a URL and outputs the three separate parts. With the URL above, for example, the output would read . . .

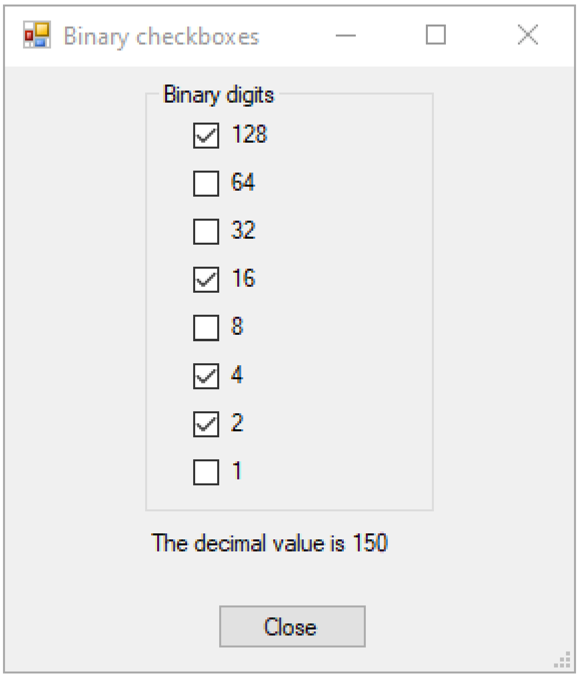
   ```
   Server: www.newcastle.edu.au
   Path: colleges/schools/sips
   File: staff.html
   ```

   Be sure that your program doesn’t crash if its input does not have all three parts.

7. The ceiling function (which you can’t ever imagine finding a use for) finds the next whole number above the number it’s given as an argument. Now, for reasons best known to herself, a customer wants your program to find the next multiple of 5 above the number it’s given as an argument. For example, 41.32, 43.0, and 44.956 would all result in the value 45.0. Work out how you would produce this result, and then test your solution in a simple program.

8. Create a form that has checkboxes for each of the bit (binary digit) values from 128 to 1.

   Whenever any of the checkboxes is checked or unchecked, the label at the bottom will automatically change to reflect the decimal value. The decimal value is simply the sum of the values of the bits; for example, 128 + 16 + 4 + 2 is 150.



Binary digits

☑ 128
☐ 64
☐ 32
☑ 16
☐ 8
☑ 4
☑ 2
☐ 1

The decimal value is 150