

COMP103 Revision

Exam-Style Revision Questions

The following questions may not bear any resemblance to past or future examination questions or cover all aspects of any exam. They relate only to course material and are for revision purposes only. To prepare fully for the exam you should review all examinable aspects of the course — lectures, tutorials, lab work, the text book, the theory test and notes you have made. We will not provide model answers to these questions.

Steve, a friend of yours, wants a small C# application built to keep track of who he loans things to. It needs to store what each person has borrowed, when and what their phone number is, so he can ring them to get it back. He has made a start at the application, but he dropped out of 103 and so has got stuck, and has asked you to help finish the program.

The program will use a data file with the following information: item, name, phone, date and whether it was returned, e.g.

```
Pen,Bob Jones,555-1212,08/02/2002,False  
Coat,Mary,025-555-999,21/11/1999,True
```

Here is Steve's current form interface:

The screenshot shows a Windows application window titled "Items Borrowed from Steve". The window has a menu bar with "Open File", "Save File", "mnuExit", and "New Item". The main area is titled "Item Borrowed:" and contains several input fields and buttons. At the top is a search field with a "Find" button. Below it is a label "please type in the name of the person the item was loaned to" next to an input field. To the right of this is a "Report by Person" button and a checkbox labeled "Returned". Below the name field is a label "lbpn" next to another input field. Further down is a label "DATE BORROWED!" next to an input field, and to its right is a "Days" input field and a "Calc Length Borrowed" button. At the bottom is a "Results Display" area with a large empty box. On the left side of the "Results Display" area are two labels: "List of Items Currently Borrowed" and "List of Items Borrowed". An "Exit" button is in the bottom right corner.

1. Indicate on the above screen shot the order of tab index you would assign, what access keys you would use and what tool tips you would add.
2. Explain what each line in the mnuOpenFile_Click event does:

```
private void mnuOpenFile_Click(object sender, EventArgs e)
{
    Clear();

    try
    {
        openFileDialog1.Filter = "Data files (*.dat)|*.dat|" +
            "All files (*.*)|*.*";
        openFileDialog1.FileName = "";
        if (openFileDialog1.ShowDialog() == DialogResult.OK)
        {
            infile = File.OpenText(openFileDialog1.FileName);
            if (infile.Peek() != -1)
            {
                frmForm1.Text = "Items borrowed from " +
                    openFileDialog1.FileName;
            }
        }
    }
    catch(Exception error)
    {
        MessageBox.Show(error.Message, "Error Warning",
            MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}
```

3. Write code for the “List of Items Currently Borrowed” command button click event. It should read from a file that is already opened in variable infile, each of the items in the file. It should add to the Results Display list box, called lstResults, any items that are currently borrowed (i.e. have not been returned). Use the following format for the list box entry:

Pen to Bob Jones (555-1212) on 08/02/2002

4. Steve admits to being confused over the following terms:
 - i) Function,
 - ii) Procedure,
 - iii) Event Handler,
 - iv) Method,
 - v) Object-Oriented Event-Driven Programming Paradigm.
 Define each term and give an example.

5. Describe why, from a GUI design point of view, Steve's current form interface is bad. Show on the form below how you could improve his design.

6. Steve is not sure if he wants the results section displayed on the same form or not. Discuss whether the results should be on the same form, or in a modal dialogue box, or in a second modeless form. What are the advantages and disadvantages of each.
7. Steve wants the Find Item command button to only be enabled when an item is typed in the txtItem text box. The text box control has the following events:
- i) Change,
 - ii) Click,
 - iii) Drag Over,
 - iv) Got Focus,
 - v) Lost Focus,
 - vi) Key Press.

Explain when each event is triggered and which would be best to use to code the enable action.

8. Write a ConvertDate procedure to be used in the following code. Use the Left, Mid and Right string processing functions to break up the date given in the format “dd/mm/yyyy” into days, months and years. For each argument to this procedure, state whether the argument should be BYVAL or BYREF.

```
private void ConvertDate(string strDate, int intDay, int
intMonth, int intYear)
{

}

private void btnCalcLengthBorrowed_Click(object sender,
EventArgs e)
{
    int intDays, intMonths, intYears;
    int intCurrentTotalDays, intBorrowTotalDays;
    int intLengthBorrowed;
    ConvertDate(txtCurrentDate, intDays, intMonths, intYears);
    intCurrentTotalDays = 365 * intYears + 30 * intMonths + intDays;
    ConvertDate(txtBorrowDate, intDays, intMonths, intYears);
    intBorrowTotalDays = 365 * intYears + 30 * intMonths + intDays;
    intLengthBorrowed = intCurrentTotalDays - intBorrowTotalDays;
    lblLength.Text = intLengthBorrowed.ToString() & " Days";
    if (intLengthBorrowed > 30)
        lblLength.ForeColor = &HFF;
    else
        if (intLengthBorrowed > 365)
            lblLength.ForeColor = &HFFFF;
}
```

9. Rewrite the btnCalcLengthBorrowed_Click procedure using appropriate constants, also indent and comment the code.
10. Steve finds that with the Calc Length Borrowed command, items borrowed over a year do not change to red like they are supposed to. Describe how the Debugger tool could be used to help find the error. What is the bug in the code, and how can he fix it.
11. Steve wonders if storing the information in a list would be better than always reading from a data file. Design a struct to hold the same information as the file, showing what fields are required and their type. Write example code to create and initialise a list of your structs.