ASSIGNMENT #1

The assignment should be presented in email attachment format (the solution – excluding the bin folder - to be packed in a zip file and attached to an email sent to me: sorin.butnaru@seneca...). The student’s name and ID should appear as a program comment. Do not forget to comment your programs. The ability of the student to design the forms and classes following the instructions is highly desired.

**(100 marks) The exercise focuses on creating a full functional application using Windows Forms**

* (**10 marks**) Create a form named **F1** to incorporate the following Controls:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Control** | **Name** | **Purpose** | **Visible at start** |
| 1 | Button | btnExit | End of application | Yes |
| 2 | Button | btnStart | Launch the second form | No |
| 3 | Button | btnAdd | Add info to Listbox | No |
| 4 | Button | btnDel | Remove row from Listbox | No |
| 5 | Listbox | lbPersons | Collect information | Yes |
| 6 | Label | lblName | Explain the textbox | Yes |
| 7 | Label | lblPhone | Explain the textbox | Yes |
| 8 | TextBox | txtName | To add one name | Yes |
| 9 | TextBox | txtPhone | To add phone number | Yes |
| 10 | CheckBox | ckBkg | If a change of the BkgColor of the second form is needed | Yes |
| 11 | ComboBox | comboCols | Select one of the colors | No |
| 12 | Label | lblErrors | Show current error | No |

* (**5 marks**) Create a form named **F2** to incorporate the following Controls:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Control** | **Name** | **Purpose** | **Visible at start** |
| 1 | Button | btnSave | Save the list to a file and exit from this form | Yes |
| 2 | Button | btnCancel | Abandon this form | Yes |
| 3 | Listbox | listBoxReceive | Collect the data we want to save | Yes |
| 4 | Label | lblMessage | Present current error | No |

**General Behaviour:**

The user is asked to enter one or multiple name(s) and phone number(s) – one by one - of some person(s) (using the textboxes 8 and 9). The name should not contain spaces (is one word). Each name+phone can be added to the list presented in the **lbPersons** by pressing the button **btnAdd.** The information is a string composed by name+space+phone (on the same row). If some data is wrong, the user can delete the row in **lbPersons** by pressing **btnDel** (the row is supposed to be selected first).

When all the desired items are entered, the content should be transferred to the listbox of the second form (“**F2**”) by pressing the **btnStart.** The transfer will be done using a list. The application can be terminated by pressing button **btnExit**.

On the second form the user can cancel the process and return to the main form (by pressing button **btnCancel**) or can start a process of updating the information located on the file “**MyPersons.txt**” (re-write the file.) After updating, the control is returned back to the main form, where the program ends.

**(55 marks**) **Functionality** (for **F1**):

* c(**5 marks**) Exit application by pressing **btnExit**
* d(**5 marks**) Add information from textboxes to the **lbPersons**
* e(**10 marks**) Remove any row from **lbPersons** after selecting it; if no selection - give a message (painted in red) in **lblErrors** with the explanation of the error. The message will be no more visible when some change to the textboxes is made, or a selection in the list, or if **btnAdd** is pressed.
* f(**10 marks**) Set visible the button **btnStart** and the combobox **comboCols** when the conditions are proper. This means: (a) in **lbPersons** minim one person exists, (b) if **ckBkg** is checked some color is selected; ***Warning***: the user has the choice NOT to change the background and then the **comboCols** should be not visible (of course, with nothing selected inside)
* g(**10 marks**) Set visible the button **btnAdd** when information in the textbox exists and hide it when the textbox is empty; set visible the button **btnDel** when there is a minim one row added to **lbPersons**
* h(**10 marks**) Load the **F2** by pressing **btnStart.** The new form will be shown as a ***modal dialog*** and will receive a list (**xList**) created from **lbPersons** in order to transfer data.
* i**(5 marks**) Set the background of **F2**, if this option was selected

**(45 marks**) **Functionality** (for **F2**):

* k(**5 marks**) Abandon the form when pressing **btnCancel** (no changes)
* l(**5 marks**) Create **myL,** a list (of type string)
* m(**10 marks**) Open the file named **MyPersons.txt** (if possible) and read each row into **myL.** If the file does not exist, **myL** will be empty. Close the file.
* n(**10 marks**) Add to **myL** the content of the **xList, but only the new records**. You need to compare each record with the existing elements of **myL.**
* o(**5 marks**) Populate the content of the listbox **ListBoxReceive** from **myL** to allow the user to see the result
* p(**10 marks**) Write the list **myL** to the text file **MyPersons.txt** and close the form (and the application, in F1) by pressing **btnSave**.

Objectives of the assignment:

1. To make the student comfortable with Visual Studio
2. To make the student comfortable with Visual Studio Graphic User Interface
3. To make the student comfortable with Visual Studio Debugging System
4. To make sure the student knows how to create a project
5. To make sure the student knows how to build a form
6. To make sure the student knows how to add a second form to the project
7. To make sure the student knows how to read/handle data (located on the hard-disk)
8. To make sure the student knows how to add simple controls to the form and set some of the properties (font, background/foreground color, size)
9. To make sure the student knows how to add data to listboxes or comboboxes
10. To make sure the student knows how to set controls visible/hidden
11. To make sure the student knows how to create buttons and add events (OnClick…)
12. To make sure the student knows how to add events to other controls (OnChange…)
13. To make sure the student knows how to create an instance of a class (for the second form)
14. To make sure the student knows to create Lists of objects
15. To make sure the student knows to open a modal dialog
16. To make sure the student knows how to pass data between the two classes/forms
17. To make sure the student knows how to write data to the disk
18. To check if the student knows how to avoid exceptions by using try/catch