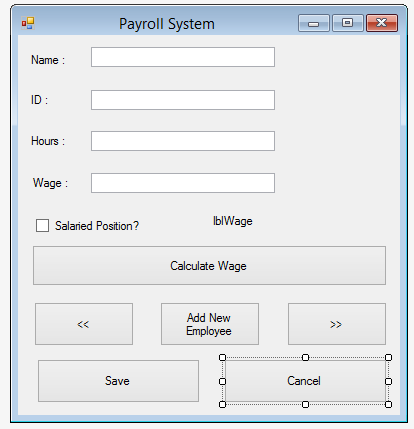
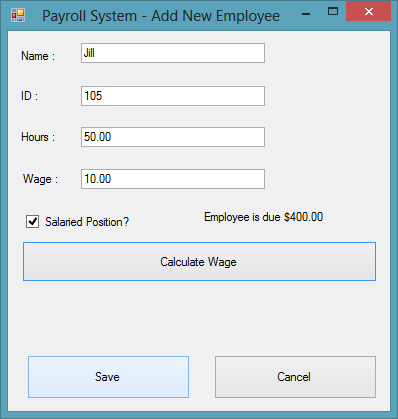
***CIS 311 Assignment 1***

For your first assignment, you will be using pretty much all of the skills that you should have learned in CS 146. I want you to write a Windows Forms based Payroll Management System that also allows the information stored in the system to be persisted out to a file between program executions. In addition, the system should allow a user to add new employees to it (you do not have to worry about allowing editing or deleting of existing employee information). Once an employee is added, they stay added. Here’s what the initial form will look like:

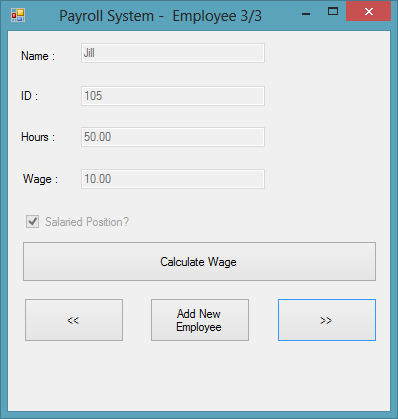


The top four items (and Salaried Position) should be pretty obvious – this is what you will need to save out to the text file. You will not store the calculated wage – that needs to be done on the fly. If an employee is salaried, you simply multiply 40.0 hours \* wage, regardless of how many hours the employee actually worked. If the employee is hourly, then you will need to accommodate for overtime.

When you start up the system for the first time, everything will be empty, so you will want to force adding a new employee since there isn’t anyone in the system yet:



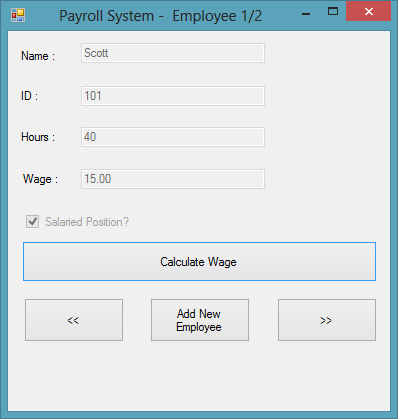
Notice that anytime the Add New Employee button has been pressed only the Calculate Wage, Save and Cancel buttons are available. Also, notice that the title has changed on the form, indicating we are in the process of adding a new employee. The textboxes and checkbox are all available and can be changed however we like. The user may press the Calculate Wage button at any time and see the wage displayed in the label halfway down the screen. Once the user has typed in the necessary information, he or she may either Save or Cancel the data addition. If the user decides to add the person, we are taken to a screen looking like this:



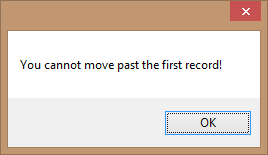
This was actually the third employee I added… Notice that the textboxes and checkbox are not editable other than when adding a new employee. At this point, we can navigate through the other records using the << and >> buttons. We can also calculate the current record’s wage as well. The Add New Employee button is there if we need to add another employee. Finally, we see that the title tells us which record we are currently viewing (current record number / count of records).

Your system needs to persist the records out to a file (you pick what you want this called – just make sure that it lives in the same directory as your project’s executable files -- I don’t want pathing to be a problem).

Here’s what things look like when I start my application back up (assuming I cancelled adding the third new employee above):



You need to add MessageBoxes if the user tries to navigate past either end of the records. A MessageBox should also be displayed if adding an employee is cancelled. You should also provide a MessageBox with an error message if the wage cannot be calculated for any reason (characters instead of numbers or missing data). In other words, this should be a well behaved application… Here’s one of my error MessageBoxes:



The easiest way to implement this would probably be to use a backing memory store like a one-dimensional array of structures or strings. You could then put each person into an element and simply read and write the array elements out to the file.

You will need to zip all of your project files (everything in the project folder and all folders/contents beneath it). You will also need to create a single Word document that contains your program’s cover sheet, source code and screenshots of the program in execution. Place this Word document in the root of the zip file and upload it to Canvas. This will be standard practice for every assignment in this class, so make sure you do things correctly or I will deduct points.