

Program 2 - EXTENSION

Worth: 75 points

Due: Thursday, October 17 (by 11:59 PM)

Purpose: This assignment explores the use of decision making logic and the creation of a GUI application.

Create a Windows Forms GUI application that will determine the earliest time that a continuing UofL *undergraduate* student may register for Spring 2020 courses using the priority registration schedule available from the Registrar's site: [Spring 2020 Priority Registration Schedule for Continuing Students](#).

Your form will need to gather the first letter of the user's last name and the user's credit hours (both from TextBoxes) to determining class standing classification. UofL does offer partial credit hour classes, so this value must be stored using one of the floating-point types. Remember, the classification is as follows: 90+ credit hours is a senior, 60+ credit hours is a junior, 30+ credit hours is a sophomore, less than that is a freshman.

The Registrar bases the registration decision on the combination class standing and first letter of the student's last name. You may either use a string with one letter or an individual **char** to store the letter. Using the **char** data type is actually easier. See p. 74-76 for the **char** data type. Unlike strings, **char** variables may be compared directly using <, <=, >, and >= operators. If you want the user to enter their entire last name, the text shows how to easily extract the first letter as a **char** in the example starting at the bottom of p. 227 of the text. Your code should not care whether the user enters their name with a capital first letter or lower case letter. You can assume that the user won't put anything other than a letter as their last name first letter, though it is not difficult to test this (see method [char.IsLetter](#), for example). Methods [char.ToUpper](#) or [char.ToLower](#) may also prove useful.

When the user submits their class standing and last name letter, your application should display the earliest date and time that the student will be allowed to register. You may use labels, textboxes, or a dialog box (as with method **MessageBox.Show**) to display the earliest registration date and time.

For this assignment, I want you to write decision logic using Chapter 4 material. We will solve this assignment a second time later, using arrays and loops. Think carefully about how you should structure your decision logic. There are many way to make this decision but some approaches are much easier and efficient than others. Think about several alternatives before committing to one. You will probably want to break this problem down into smaller steps and build the application in pieces. For example, designing the layout of the GUI as one step. Adding support for seniors as another step, then adding juniors, etc. As you consider the design of your decision logic, look for opportunities to reduce duplication of code by separating the identification of the earliest date from the identification of the earliest time. Note that there are really only two patterns for the time of registration, one pattern for juniors and seniors and one pattern for freshmen and sophomores. Take advantage of this to reduce the size of the code you must write. Strive for efficiency with your decision logic when testing letter ranges, as well. Don't test both ends of a range if you can avoid it.

Be sure to add appropriate comments in your code, including your **Grading ID** (not name nor student ID), program number, due date, and course section. Each variable used in your program needs a comment describing its purpose. These requirements are expected for every program and are listed in the syllabus. Preconditions and postconditions are not expected yet, as we've not covered them in class. However, comments are expected for important sections of code, including event handlers.

As with our labs, I'm asking you to upload a compressed ZIP archive of the entire project. Rather than giving me floppy disks or printouts, you will upload **all your files** to Blackboard using the *Assignments* tool. I'm asking you to upload a compressed ZIP archive of the entire project, just as with our labs. The steps for doing this will vary somewhat based on the ZIP utility being used. Before you upload this .ZIP file, it's a good idea to make sure that everything was properly zipped. Make sure your code is present and you can run your file. Once you have verified everything, return to the *Assignments, Programs* area of Blackboard. Click on "Program 2" and the *Upload Assignment* page will appear. Add any

comments you like in *Comments* field. Click *Browse* next to *File to Attach* to browse the system for your file. Browse to the location of your .ZIP file and select it. Note, multiple files may be attached using the *Add Another File* option. For this assignment, we just need the "Prog2.zip" file. Make sure everything is correct in the form and then click *Submit* to complete the assignment and upload your file to be graded.

Remember, this is an **individual** assignment. Please be mindful of the syllabus' statement on academic dishonesty. If you are unsure about what constitutes academic dishonesty, **ASK!**