```
// Program 3
// CIS 199-02
// Due: 4/13/2017
// Grading ID: B3049

// This application calculates the earliest date in which a student at UofL is able to
register for the Fall
// semester of 2017. Uses the first letter of someones last name and academic standing
to calculate the exact date and
// time in which you are able to register.

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace Prog2
{
    public partial class RegForm : Form
    {
        public RegForm()
        {
            InitializeComponent();
        }

        // Find and display earliest registration time
        // Preconditions: Code must compile without errors in order to click button
        // Postconditions: Click button executes code
        private void findRegTimeBtn_Click(object sender, EventArgs e)
        {
            const string DAY1 = "March 29";  // 1st day of registration
            const string DAY2 = "March 30";  // 2nd day of registration
            const string DAY3 = "March 31";  // 3rd day of registration
            const string DAY4 = "April 3";   // 4th day of registration
            const string DAY5 = "April 4";   // 5th day of registration
            const string DAY6 = "April 5";   // 6th day of registration

            const string TIME1 = "8:30 AM";  // 1st time block
            const string TIME2 = "10:00 AM"; // 2nd time block
            const string TIME3 = "11:30 AM"; // 3rd time block
            const string TIME4 = "2:00 PM";  // 4th time block
            const string TIME5 = "4:00 PM";  // 5th time block

            string lastNameStr;       // Entered last name
            char lastNameLetterCh;    // First letter of last name, as char
            string dateStr = "Error"; // Holds date of registration
            string timeStr = "Error"; // Holds time of registration
            bool isUpperClass;        // Upperclass or not?

            string[] lowerclassmanTimes = { TIME5, TIME1, TIME2, TIME3, TIME4, TIME5,
TIME1, TIME2, TIME3, TIME4 }; // Array with times that correspond with lastNameCharArray
            string[] upperclassmanTimes = { TIME3, TIME3, TIME4, TIME4, TIME5, TIME5,
TIME1, TIME2, TIME3, TIME4 }; // Array with times that correspond with lastNameCharArray
```

```
            char[] lastNameCharArray = { 'A', 'C', 'E', 'G', 'J', 'M', 'P', 'S', 'U', 'W'
}; // Array of last name first letters

            lastNameStr = lastNameTxt.Text;
            if (lastNameStr.Length > 0) // Empty string?
            {
                lastNameLetterCh = lastNameStr[0];   // First char of last name
                lastNameLetterCh = char.ToUpper(lastNameLetterCh); // Ensure upper case

                if (char.IsLetter(lastNameLetterCh)) // Is it a letter?
                {
                    isUpperClass = (seniorRBtn.Checked || juniorRBtn.Checked);

                    // Juniors and Seniors share same schedule but different days
                    if (isUpperClass)
                    {
                        if (seniorRBtn.Checked)
                            dateStr = DAY1;
                        else // Must be juniors
                            dateStr = DAY2;

                        int index = lastNameCharArray.Length - 1;
                        while (index >= 0 && lastNameLetterCh < lastNameCharArray[index])
                            index--;

                        timeStr = upperclassmanTimes[index];
                    }
                    // Sophomores and Freshmen
                    else // Must be soph/fresh
                    {
                        if (sophomoreRBtn.Checked)
                        {
                            // C-O on one day
                            if ((lastNameLetterCh >= 'C') && // >= C and
                                (lastNameLetterCh <= 'O'))   // <= O
                                dateStr = DAY4;
                            else // All other letters on previous day
                                dateStr = DAY3;
                        }
                        else // must be freshman
                        {
                            // C-O on one day
                            if ((lastNameLetterCh >= 'C') && // >= C and
                                (lastNameLetterCh <= 'O'))   // <= O
                                dateStr = DAY6;
                            else // All other letters on previous day
                                dateStr = DAY5;
                        }

                        int index = lastNameCharArray.Length - 1;
                        while (index >= 0 && lastNameLetterCh < lastNameCharArray[index])
                            index--;

                        timeStr = lowerclassmanTimes[index];
                    }

                    // Output results
                    dateTimeLbl.Text = dateStr + " at " + timeStr;
```

```
                }
                else // First char not a letter
                    MessageBox.Show("Make sure last name starts with a letter");
            }
            else // Empty textbox
                MessageBox.Show("Enter a last name!");
        }
    }
}
```