

```
1 // Program 2
2 // Grading ID: B3051
3 // CIS 199-75
4 // Due: 11/9/2017
5
6 // This application calculates the earliest registration date
7 // and time for an undergraduate student given their class standing
8 // and last name.
9 // Decisions based on UofL Spring 2018 Priority Registration Schedule
10
11 using System;
12 using System.Collections.Generic;
13 using System.ComponentModel;
14 using System.Data;
15 using System.Drawing;
16 using System.Linq;
17 using System.Text;
18 using System.Windows.Forms;
19
20 namespace Prog3
21 {
22     public partial class RegForm : Form
23     {
24         public RegForm()
25         {
26             InitializeComponent();
27         }
28
29         // Find and display earliest registration time
30         private void findRegTimeBtn_Click(object sender, EventArgs e)
31         {
32             const string DAY1 = "November 3"; // 1st day of registration
33             const string DAY2 = "November 6"; // 2nd day of registration
34             const string DAY3 = "November 7"; // 3rd day of registration
35             const string DAY4 = "November 8"; // 4th day of registration
36             const string DAY5 = "November 9"; // 5th day of registration
37             const string DAY6 = "November 10"; // 6th day of registration
38
39             const string TIME1 = "8:30 AM"; // 1st time block
40             const string TIME2 = "10:00 AM"; // 2nd time block
41             const string TIME3 = "11:30 AM"; // 3rd time block
42             const string TIME4 = "2:00 PM"; // 4th time block
43             const string TIME5 = "4:00 PM"; // 5th time block
44
45             char[] SraJrletters = { 'D', 'I', 'O', 'S', 'Z' }; //Initializes the ↗
46             string[] SraJrtimes = { TIME2, TIME3, TIME4, TIME5, TIME1 }; // ↗
47             //Initializes the time array for Jr and Sr
```

```
48     char[] SoaFrletters = { 'C', 'D', 'F', 'I', 'L', 'O', 'Q', 'S', 'V', 'Z' }; //Initializes the character array for Fr and So
49     string[] SoaFrtimes = { TIME3, TIME4, TIME5, TIME1, TIME2, TIME3, TIME4, TIME5, TIME1, TIME2 }; //Initializes the time array for Fr
    and So
50
51     string lastNameStr; // Entered last name
52     char lastNameLetterCh; // First letter of last name, as char
53     string dateStr = "Error"; // Holds date of registration
54     string timeStr = "Error"; // Holds time of registration
55     bool isUpperClass; // Upperclass or not?
56     bool istancefound = false; //states if time is found
57
58     //This section gets the date
59     lastNameStr = lastNameTxt.Text;
60     if (lastNameStr.Length > 0) // Empty string?
61     {
62         lastNameLetterCh = lastNameStr[0]; // First char of last name
63         lastNameLetterCh = char.ToUpper(lastNameLetterCh); // Ensure
        upper case
64
65         if (char.IsLetter(lastNameLetterCh)) // Is it a letter?
66         {
67             isUpperClass = (seniorRBtn.Checked || juniorRBtn.Checked);
68
69             // Juniors and Seniors share same schedule but different days
70             if (isUpperClass)
71             {
72                 if (seniorRBtn.Checked)
73                     dateStr = DAY1;
74                 else // Must be juniors
75                     dateStr = DAY2;
76
77                 for(int a = 0; a<SraJrletters.Length && !istancefound; a+
78 +) //searches for time
79                 {
80                     if(SraJrletters[a]>= lastNameLetterCh)
81                     {
82                         istancefound = true;
83                         timeStr = SraJrtimes[a];
84                         break;
85                     }
86                 }
87             }
88             // Sophomores and Freshmen
89             else // Must be soph/fresh
90             {
91                 if (sophomoreRBtn.Checked)
```

```
92     {
93         // G-S on one day
94         if ((lastNameLetterCh >= 'G') && // >= G and
95             (lastNameLetterCh <= 'S')) // <= S
96             dateStr = DAY4;
97         else // All other letters on previous day
98             dateStr = DAY3;
99     }
100     else // must be freshman
101     {
102         // G-S on one day
103         if ((lastNameLetterCh >= 'G') && // >= G and
104             (lastNameLetterCh <= 'S')) // <= S
105             dateStr = DAY6;
106         else // All other letters on previous day
107             dateStr = DAY5;
108     }
109
110     for (int a = 0; a < SoaFrletters.Length && !istimefound; a++) //Searches for time
111     {
112         if (SoaFrletters[a] >= lastNameLetterCh)
113         {
114             istimefound = true;
115             timeStr = SoaFrTimes[a];
116             break;
117         }
118     }
119 }
120
121 // Output results
122 dateTimeLbl.Text = dateStr + " at " + timeStr;
123 }
124 else // First char not a letter
125     MessageBox.Show("Make sure last name starts with a letter");
126 }
127 else // Empty textbox
128     MessageBox.Show("Enter a last name!");
129 }
130 }
131 }
132 }
```