```
    InitializeComponent();
}
// This solution keeps the first letter of the last name as a char
// and uses if/else logic for the times.
// It uses defined strings for the dates and times to make it easier
// to maintain.
// This solution takes advantage of the fact that there really are
// only two different time patterns used. One for juniors and seniors
// and one for sophomores and freshmen. The pattern for sophomores
// and freshmen is complicated by the fact the certain letter ranges
// get one date and other letter ranges get another date.

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();
        }

        //Preconditions: method must be passed a char and a char array for comparison
        //Postconditions: method will return the subscript of the matching char range in the array

        private string GetRegTime(char lastNameLetterCh, char [] lastNameLetter, string [] time )
        {
            bool found;               //boolean for stepping through array
            found = false;            //initialize boolean
            int index;                //variable to hold subscript
            index = 0;                //initialize index variable

            //while loop to step throught the array for range matching
            while (index < lastNameLetter.Length && !found)
            {
                //if range matches
                if (lastNameLetterCh <= lastNameLetter[index])
                {
                    found = true;
                }
```

```
                else //move to next
                {
                    index++;
                }
            }
            //return time at index position of match
            return time[index];
        }

        //Preconditions: event handler requires number of hours>=0 and a last name to be entered into
textbox
        //Postconditions: will return registration date and time based on number of completed hours and
first letter of last name

        private void findRegTimeBtn_Click(object sender, EventArgs e)
        {
            const float SENIOR_HOURS = 90;    // Min hours for Senior
            const float JUNIOR_HOURS = 60;    // Min hours for Junior
            const float SOPHOMORE_HOURS = 30; // Min hours for Soph.

            const string DAY1 = "November 4";  // 1st day of registration
            const string DAY2 = "November 7";  // 2nd day of registration
            const string DAY3 = "November 9";  // 3rd day of registration
            const string DAY4 = "November 10"; // 4th day of registration
            const string DAY5 = "November 11"; // 5th day of registration
            const string DAY6 = "November 14"; // 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
            float creditHours;        // Entered credit hours

if (float.TryParse(creditHrTxt.Text, out creditHours) && creditHours >= 0) // Valid hours
{
    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?
        {
            // Juniors and Seniors share same schedule but different days
            if (creditHours >= JUNIOR_HOURS)
            {
                if (creditHours >= SENIOR_HOURS)
                    dateStr = DAY1;
```

```
                            else // Must be juniors
                                dateStr = DAY2;

                            char [] lastNameLetter={ 'D','I','O','S','Z'};            //declare and
initialze array to hold end points of last name letter categories
                            string[] time = { TIME4, TIME5, TIME1, TIME2, TIME3 };    //declare and
initialize array to hold time designations for registration

                            //get reg time from method
                            timeStr = GetRegTime(lastNameLetterCh, lastNameLetter, time);


                        }
                        // Sophomores and Freshmen
                        else // Must be soph/fresh
                        {
                            if (creditHours >= SOPHOMORE_HOURS)
                            {
                                // J-V on one day
                                if ((lastNameLetterCh >= 'J') && // >= J and
                                    (lastNameLetterCh <= 'V'))   // <= V
                                    dateStr = DAY3;
                                else // All other letters on next day
                                    dateStr = DAY4;
                            }
                            else // must be freshman
                            {
                                // J-V on one day
                                if ((lastNameLetterCh >= 'J') && // >= J and
                                    (lastNameLetterCh <= 'V'))   // <= V
                                    dateStr = DAY5;
                                else // All other letters on next day
                                    dateStr = DAY6;
                            }

                            char[] lastNameLetter = { 'B', 'D', 'F', 'I', 'L', 'O', 'Q', 'S', 'V', 'Z' };
//declare and initialze array to hold end points of last name letter categories
                            string[] time = { TIME2, TIME3, TIME4, TIME5, TIME1, TIME2, TIME3, TIME4,
TIME5, TIME1 };    //declare and initialize array to hold time designations for registration

                            //get reg time from method
                            timeStr = GetRegTime(lastNameLetterCh, lastNameLetter, time);
                        }

                        // Output results
                        dateTimeLbl.Text = dateStr + " at " + timeStr;
                    }
                    else // First char not a letter
                        MessageBox.Show("Enter valid last name!");
                }
                else // Empty textbox
                    MessageBox.Show("Enter a last name!");
            }
            else // Can't parse credit hours
                MessageBox.Show("Please enter valid credit hours earned!");
        }
    }
}
```