CST8253 Web Programming II

Lab 8

Objective

1. Use ASP.NET Validation controls
2. Understand and use polymorphism in OOP

Due Date

See Brightspace posting for the due date of this lab. To earn 5 points, you are required:

1. Complete the lab as required.
2. Zip the web application project folder and submit the zipped file to the Brightspace before due date.
3. Demo your lab work during the lab session in the week after the due date .

Preparations

In this lab, you will combine functionality in Lab 6 and Lab 7. You can reuse some of the code in the previous labs

1. In Visual Studio create a new Web Form ASP.NET Web Application.
2. Add **Course.cs** file you created in Lab 6 to the project's **Models** folder
3. Add **Student.cs**, **FulltimeStudent.cs**, **ParttimeStudent.cs** and **CoopStudent.cs** you created in Lab 7 to the project's **Models** folder
4. Modify the class **Student** as follows

   Add a new readonly property **RegisteredCourses** of type **List<Course>**. Initialize this property in the Student class' constructor to an empty **List<Course>**.

   Add a virtual method **RegisterCourses** as

   ```
   public virtual void RegisterCourses(List<Course> selectedCourses)
   {...}
   ```

   This method first removes all elements in the **RegisteredCourse** and then adds the **selectedCourses** to the **RegisteredCourse**.

Add a method **`TotalWeeklyHours`** as

```
public int TotalWeeklyHours() { ... }
```

This method returns the calculated total hours of all registered courses by the student.

5. Modify the classes **FulltimeStudent**, **ParttimeStudent** and **CoopStudent**

   Add the following method to all these three classes to override the same method define in their base class **Student**:

   ```
   public override void RegisterCourses(List<Course> selectedCourses)
   {...}
   ```

   Although the method has the same name for all three classes, they implement different business rules for **FulltimeStudent**, **ParttimeStudent** and **CoopStudent** as followings, respectively.

   - For full time students – The weekly hours of the total registered courses cannot exceed **MaxWeeklyHours** defined **FulltimeStudent** class.

   - For part time students – The total number of the courses cannot register more than **MaxNumOfCourses** defined in **ParttimeStudent**.

   - Coop students – The weekly hours of the total registered courses cannot exceed **MaxWeeklyHours** defined **CoopStudent** class. The total number of the courses cannot register more than **MaxNumOfCourses** defined in **CoopStudent**.

     If the **selectedCourses** violate any of above business rules for **FulltimeStudent**, **ParttimeStudent** and **CoopStudent**, respectively, the method throws an exception with an appropriate message. Otherwise, the method invokes the base class' **`RegisterCourses`** `to add the` **selectedCourses** to the student's registered course list.

6. Add `public override string ToString() { ... }` Method to **FulltimeStudent**, **ParttimeStudent** and **CoopStudent** classes. This method return a string in the form of:

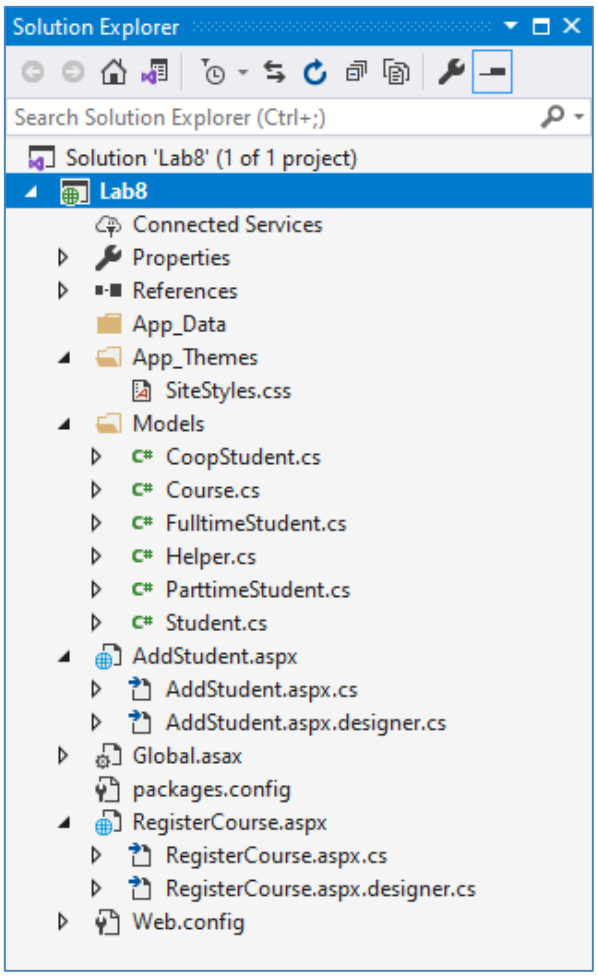
   Id – Name (*student type*)

   Where *student type* is "Full time", "Part time" or "Coop" for **FulltimeStudent**, **ParttimeStudent** and **CoopStudent**, respectively.

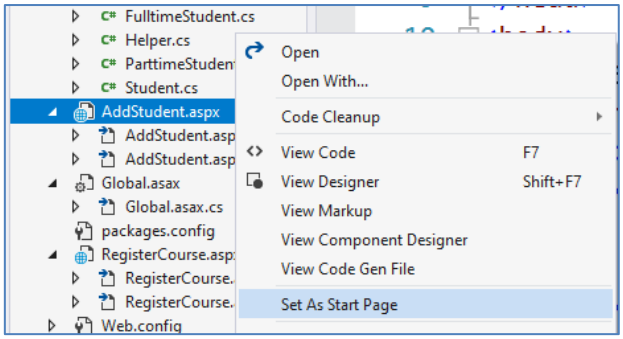
7. Add **Helper.cs** file (the same used in lab 6) to the Models folder of the project. Study the class and the methods defined in the Helper.cs file. You will use them in the rest of this lab.

8. Add the web page **AddStudent.aspx, AddStudent.aspx.cs** and **AddStudent.aspx.designer.cs** created in Lab 7 to the project.

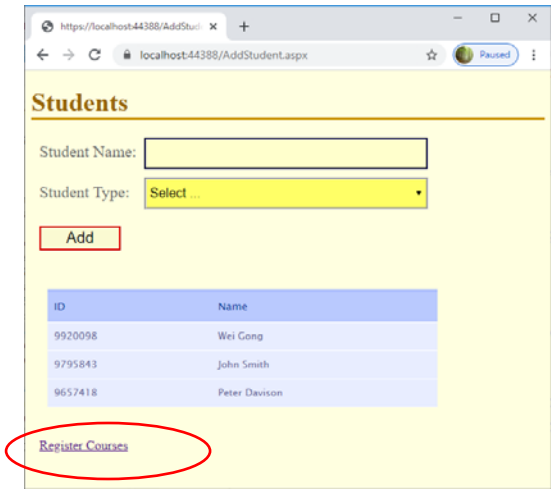
9. Add a new web page **RegisterCourse.aspx** to the project. After complete this step, the Solution Explorer of this project should look like:



Now that you have two pages in the project, you can set **AddStudent.aspx** page as the project's Start Page:



10. Add a link (HTML anchor tag) on AddStudent.aspx page. The user can navigate to RegerterCourse.aspx page by click this link.



11. Open **Global.asax.cs** add the following inside **Application_Start(...)** method. This method is only invoked once when application first starts. It is the best place to initialize static properties in the application.

```
FulltimeStudent.MaxWeeklyHours = 16;
ParttimeStudent.MaxNumOfCourses = 3;
CoopStudent.MaxNumOfCourses = 2;
CoopStudent.MaxWeeklyHours = 4;
```

12. Add the following to Web.config file's `<configuration>` for using ASP.NET validation controls on ASP.NET web pages

```
<appSettings>
 <add key="ValidationSettings:UnobtrusiveValidationMode" value="None"/>
</appSettings>
```

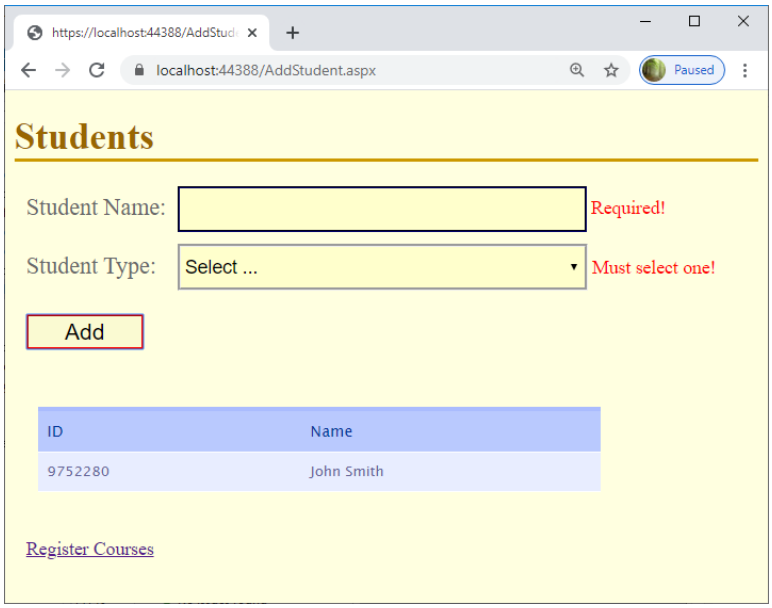
The Web.config file should look like:

```
<configuration>
  <appSettings>
    <add key="ValidationSettings:UnobtrusiveValidationMode" value="None"/>
  </appSettings>
  <system.web>
```

Requirements

1. Add validation controls to AddStudent.aspx so that both Student Name and Student Type fields are required fields, that is, the user must enter a student name and select a

student type from the dropdown list. Otherwise the error message(s) will be shown next to the field(s) as in the following screen capture:



2. **RegisterCourse.aspx** page should have following elements on the page:

- A dropdown list of all students added to the system on **AddStudent.aspx** page.

  The display text of the dropdown item should be in the following format:

  Student Id – Student Name (*Student Type*)

  Hint: Use **ToString( )** defined in step 4 to get a string in above format.

  This is a required field. If the user did not select a student before clicking the Save button, an error message will be shown next to the field.

- A registration summary showing the number of courses and total weekly hours the selected student has registered.

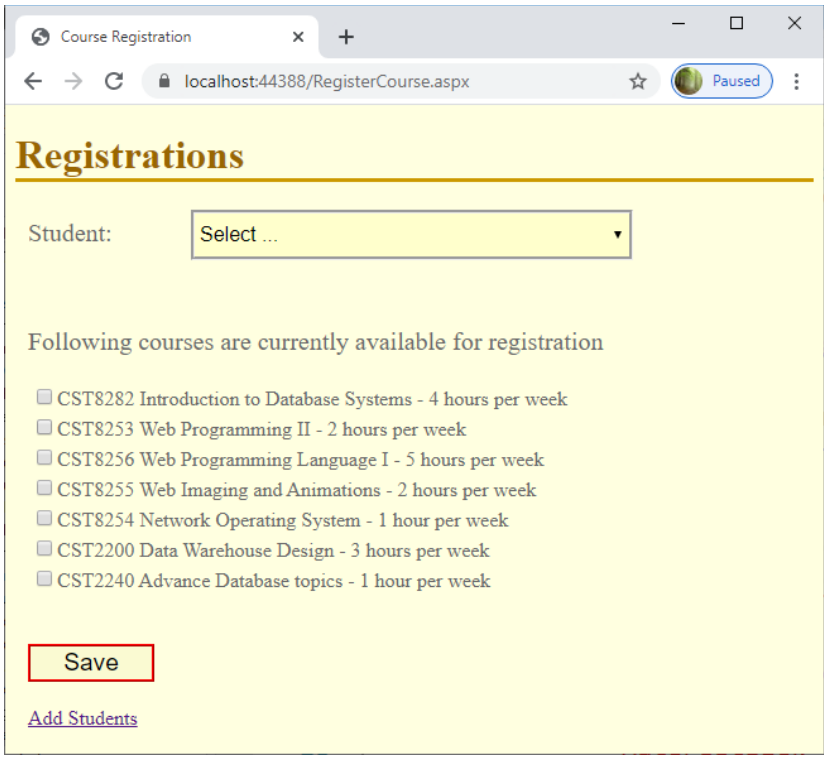
- A checkbox list for course currently available for registration.

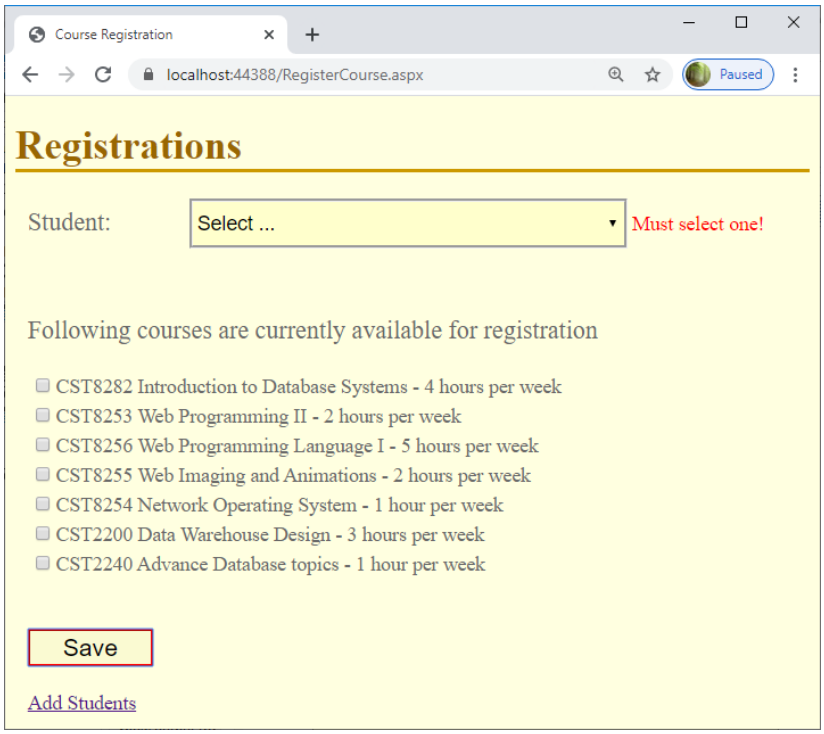
  Hint: Use `public static List<Course> GetAvailableCourses()` method of the **Helper** class to get a list of available courses.

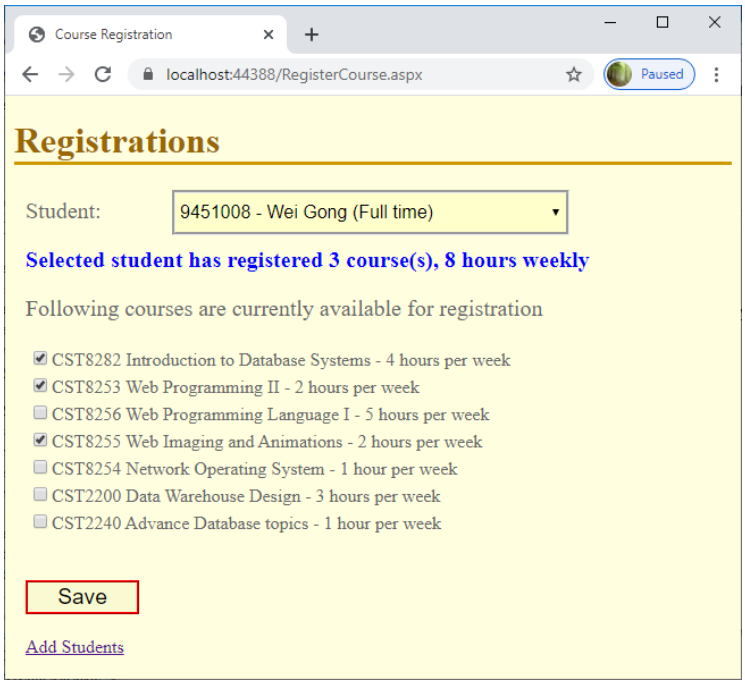
- A submit button to submit the registration.

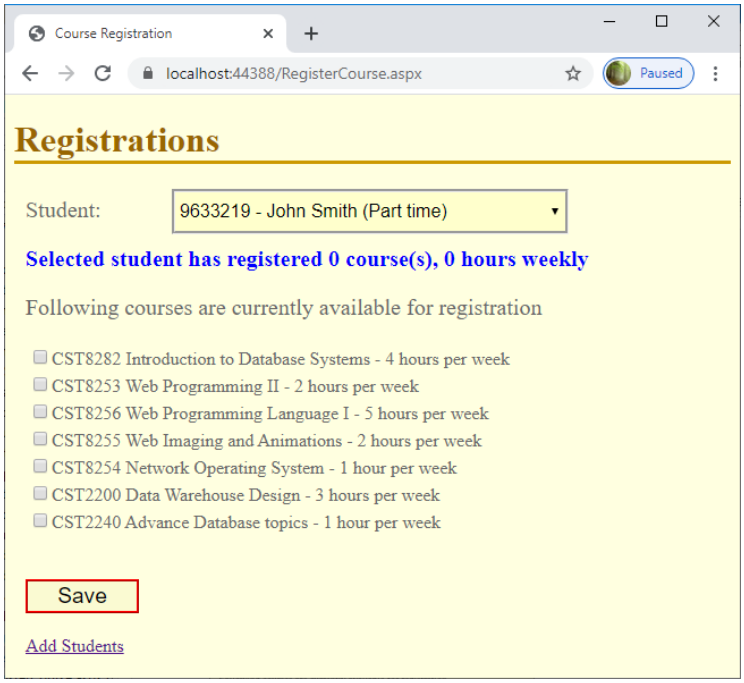
- A link which navigates the user back to **AddStudent.aspx** page.

You are required to style the web page. At minimum, all elements on the page should vertically/horizontally aligned. Up to 2 points may be deducted for un-styled or poorly styled page(s). The followings are a couple of sample screen captures of the web page:









3. **RegisterCourse.aspx** should have the following behaviors:

- After the user selected a student from the dropdown list, the web page shows the selected student's registration summary and pre-check the courses the student has already registered as shown in the second and the third screen capture in step 9 so that the user can modify the selected student's registered courses.

- The user can then check and/or un-check courses to select or de-select the courses to register.

- After the use clicked the **Save** button, if the user's selected courses violate the any of the business rules below (same as listed in step 3), the page displays an error message without change the selected student's registration:

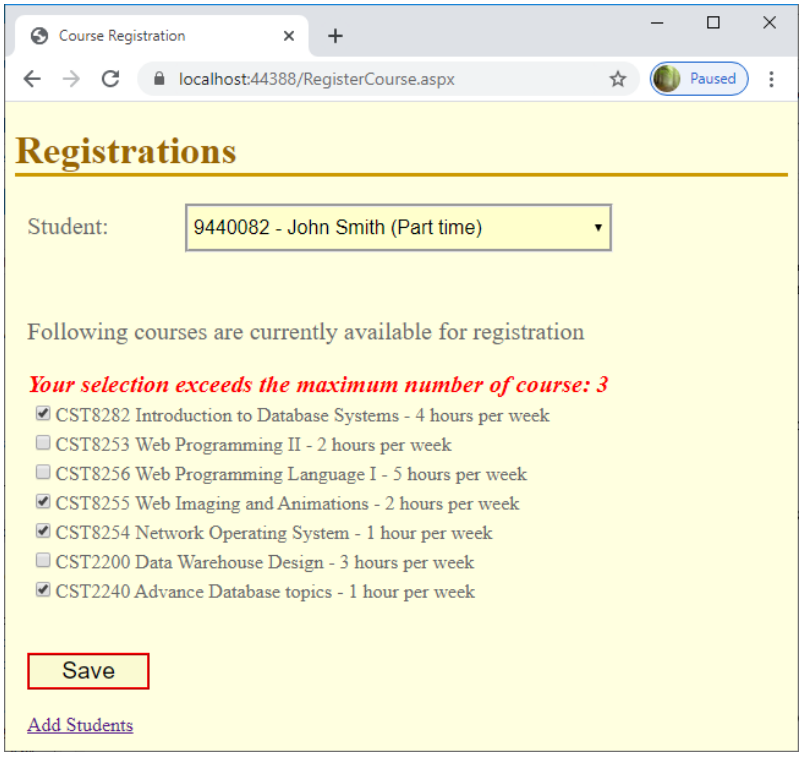
    a. For full time students – The weekly hours of the total registered courses cannot exceed **MaxWeeklyHours** defined **FulltimeStudent** class.

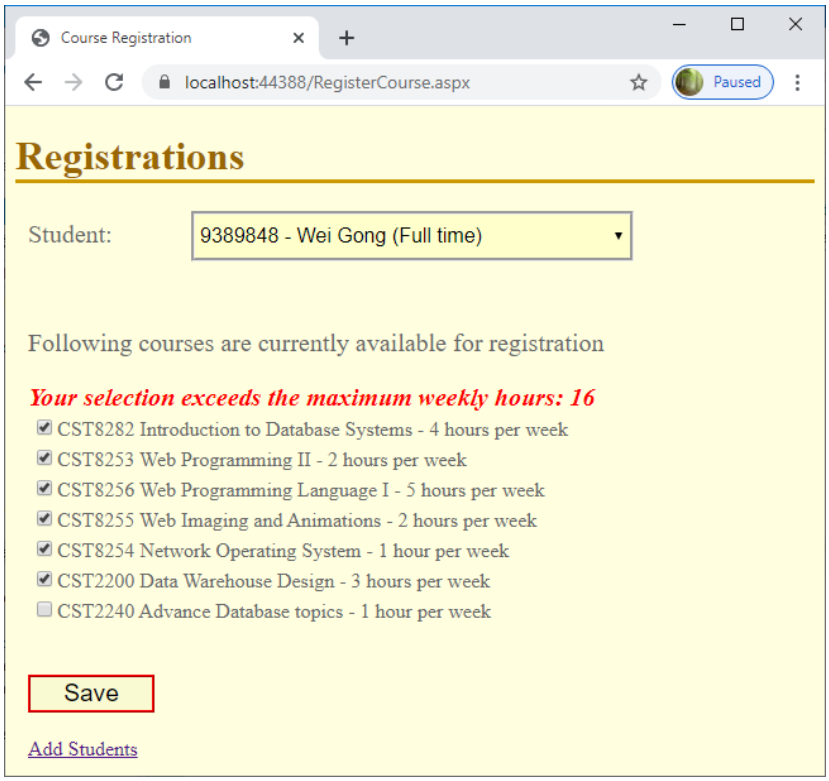
    b. For part time students – The total number of the courses cannot register more than **MaxNumOfCourses** defined in **ParttimeStudent**.

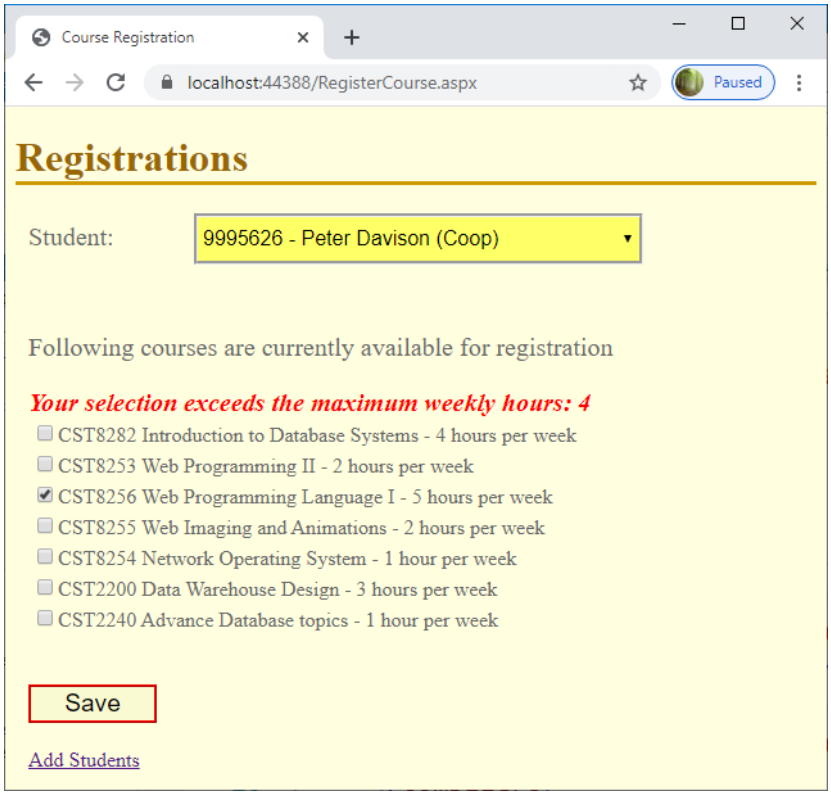
    c. Coop students – The weekly hours of the total registered courses cannot exceed **MaxWeeklyHours** defined **CoopStudent** class. The total number of the courses cannot register more than **MaxNumOfCourses** defined in **CoopStudent**.

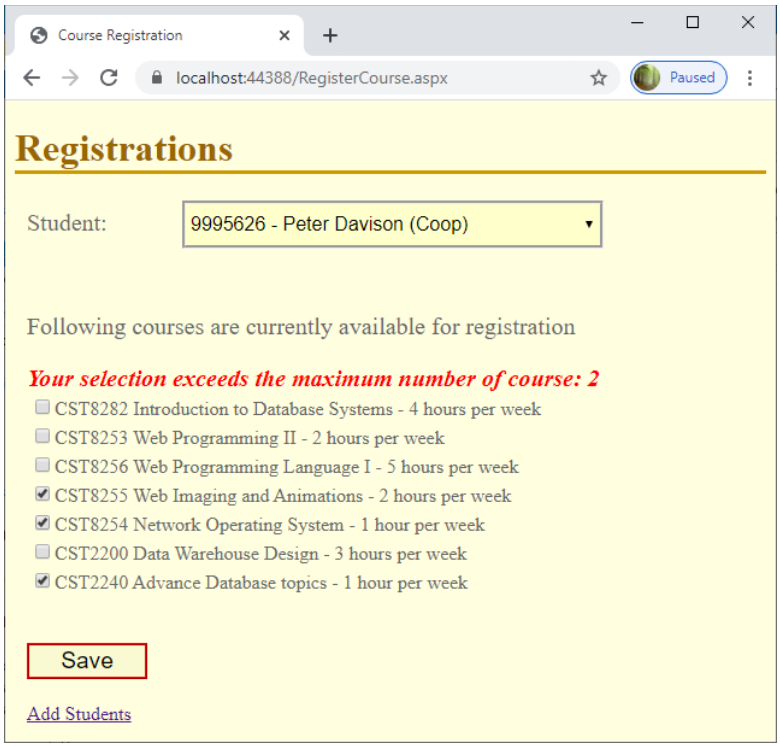
Hint: Use `public static Course GetCourseByCode(string code)` of the Helper class to get the Course object with specified code.

The followings are a couple sample screen captures of the page when the user's selection violates one or more of the business rules above:

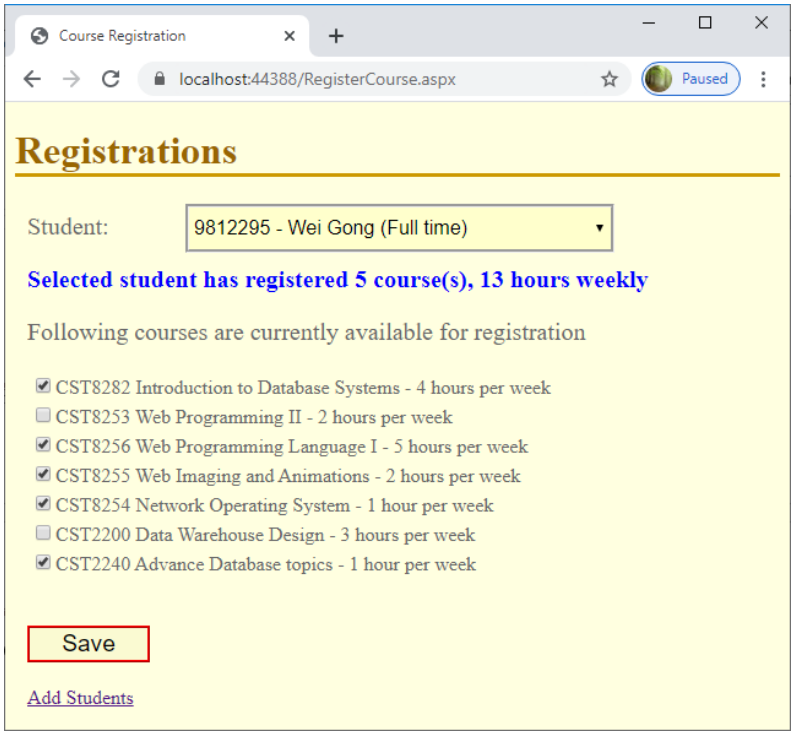
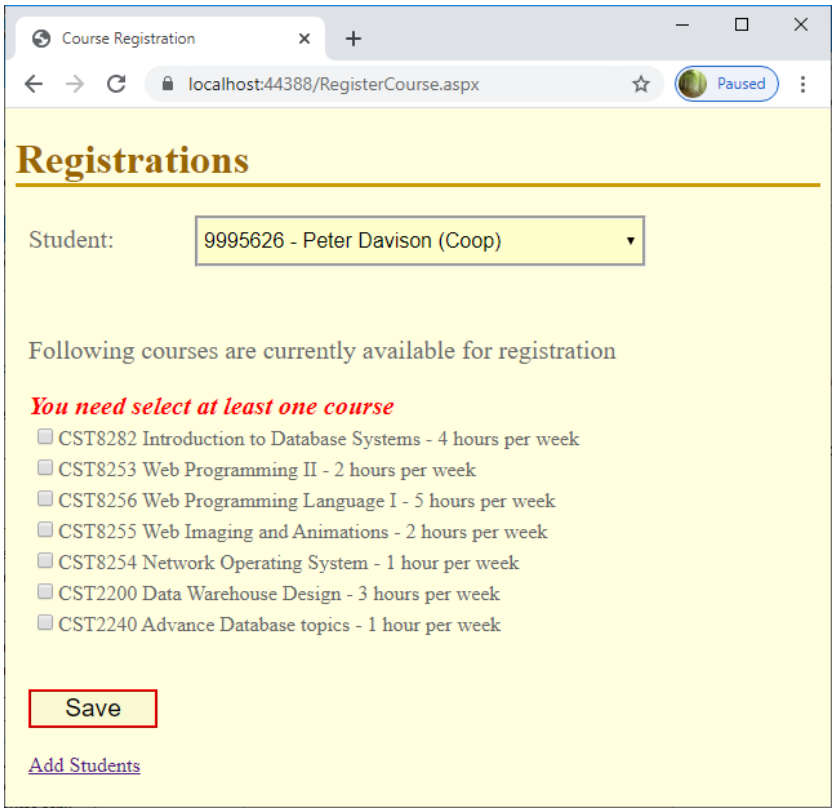



Course Registration
localhost:44388/RegisterCourse.aspx
Paused
Registrations
Student:
9389848 - Wei Gong (Full time)
Following courses are currently available for registration
Your selection exceeds the maximum weekly hours: 16
CST8282 Introduction to Database Systems - 4 hours per week
CST8253 Web Programming II - 2 hours per week
CST8256 Web Programming Language I - 5 hours per week
CST8255 Web Imaging and Animations - 2 hours per week
CST8254 Network Operating System - 1 hour per week
CST2200 Data Warehouse Design - 3 hours per week
CST2240 Advance Database topics - 1 hour per week
Save
Add Students


Course Registration
localhost:44388/RegisterCourse.aspx
Paused
Registrations
Student:
9995626 - Peter Davison (Coop)
Following courses are currently available for registration
Your selection exceeds the maximum weekly hours: 4
CST8282 Introduction to Database Systems - 4 hours per week
CST8253 Web Programming II - 2 hours per week
CST8256 Web Programming Language I - 5 hours per week
CST8255 Web Imaging and Animations - 2 hours per week
CST8254 Network Operating System - 1 hour per week
CST2200 Data Warehouse Design - 3 hours per week
CST2240 Advance Database topics - 1 hour per week
Save
Add Students



- If the user's selected courses satisfy the above listed business rules, the application registers the selected student for the selected courses and update the registration summary.



- If the user clicks the Save button when no course is checked, the application shows an error message: