

MAINFORMDESIGNER.cs

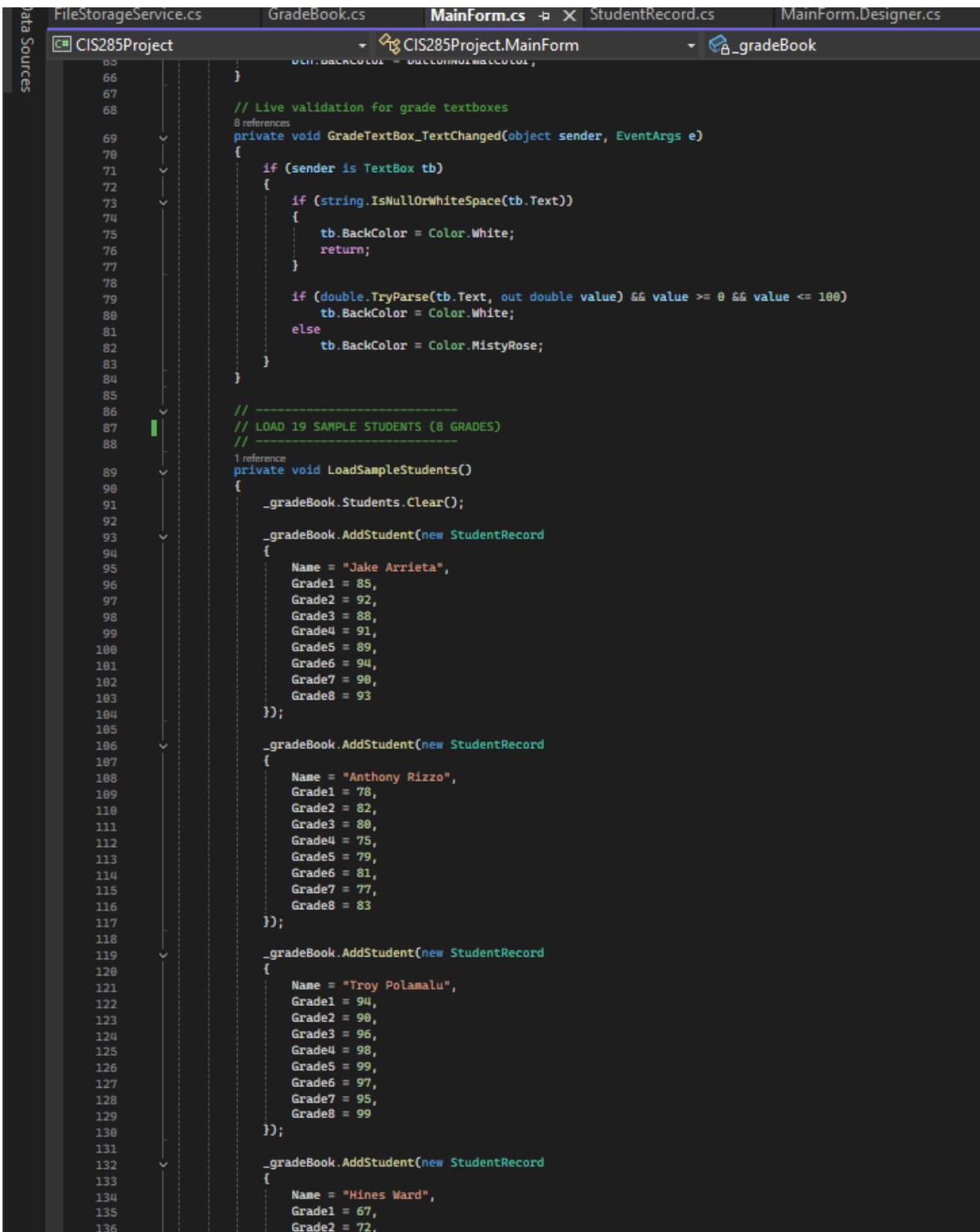
```
FileStorageService.cs      GradeBook.cs      StudentRecord.cs      MainForm.cs      MainForm.Designer.cs  ↗ CIS285Project.MainForm
CIS285Project
74 // DATAGRIDVIEW
75 this.dgvStudents.Location = new System.Drawing.Point(25, 25);
76 this.dgvStudents.Size = new System.Drawing.Size(800, 350);
77 this.dgvStudents.ReadOnly = true;
78 this.dgvStudents.RowHeadersVisible = false;
79 this.dgvStudents.SelectionMode = System.Windows.Forms.DataGridViewSelectionMode.FullRowSelect;
80 this.dgvStudents.AllowUserToAddRows = false;
81 this.dgvStudents.AllowUserToDeleteRows = false;
82 this.dgvStudents.MultiSelect = false;
83 this.dgvStudents.ScrollBars = System.Windows.Forms.ScrollBars.Both;
84 this.dgvStudents.EnableHeadersVisualStyles = false;
85 this.dgvStudents.ColumnHeadersDefaultCellStyle.BackColor =
86     System.Drawing.Color.FromArgb(0xBF, 0xA7, 0xE2);
87 this.dgvStudents.ColumnHeadersDefaultCellStyle.ForeColor =
88     System.Drawing.Color.White;
89 this.dgvStudents.BackgroundColor = System.Drawing.Color.White;
90
91 this.dgvStudents.Columns.AddRange(new System.Windows.Forms.DataGridViewColumn[] {
92     this.colName, this.colG1, this.colG2, this.colG3, this.colG4,
93     this.colG5, this.colG6, this.colG7, this.colG8, this.colAvg
94 });
95
96 this.dgvStudents.SelectionChanged += new System.EventHandler(this.dgvStudents_SelectionChanged);
97
98 // Column bindings
99 this.colName.DataPropertyName = "Name";
100 this.colG1.DataPropertyName = "Grade1";
101 this.colG2.DataPropertyName = "Grade2";
102 this.colG3.DataPropertyName = "Grade3";
103 this.colG4.DataPropertyName = "Grade4";
104 this.colG5.DataPropertyName = "Grade5";
105 this.colG6.DataPropertyName = "Grade6";
106 this.colG7.DataPropertyName = "Grade7";
107 this.colG8.DataPropertyName = "Grade8";
108 this.colAvg.DataPropertyName = "Average";
109
110 // Column headers
111 this.colName.HeaderText = "Student Name";
112 this.colName.Width = 140;
113
114 this.colG1.HeaderText = "Grade 1"; this.colG1.Width = 60;
115 this.colG2.HeaderText = "Grade 2"; this.colG2.Width = 60;
116 this.colG3.HeaderText = "Grade 3"; this.colG3.Width = 60;
117 this.colG4.HeaderText = "Grade 4"; this.colG4.Width = 60;
118 this.colG5.HeaderText = "Grade 5"; this.colG5.Width = 60;
119 this.colG6.HeaderText = "Grade 6"; this.colG6.Width = 60;
120 this.colG7.HeaderText = "Grade 7"; this.colG7.Width = 60;
121 this.colG8.HeaderText = "Grade 8"; this.colG8.Width = 60;
122
123 this.colAvg.HeaderText = "Average";
124 this.colAvg.Width = 80;
125
126 // BUTTONS
127 this.btnAdd.Location = new System.Drawing.Point(850, 25);
128 this.btnAdd.Size = new System.Drawing.Size(220, 40);
129 this.btnAdd.Text = "Add Student";
130 this.btnAdd.Click += new System.EventHandler(this.btnAdd_Click);
131 this.btnAdd.MouseEnter += new System.EventHandler(this.Button_MouseEnter);
132 this.btnAdd.MouseLeave += new System.EventHandler(this.Button_MouseLeave);
133
134 this.btnEdit.Location = new System.Drawing.Point(850, 75);
135 this.btnEdit.Size = new System.Drawing.Size(220, 40);
136 this.btnEdit.Text = "Edit Student";
137 this.btnEdit.Click += new System.EventHandler(this.btnedit_Click);
138 this.btnEdit.MouseEnter += new System.EventHandler(this.Button_MouseEnter);
139 this.btnEdit.MouseLeave += new System.EventHandler(this.Button_MouseLeave);
140
141 this.btnDelete.Location = new System.Drawing.Point(850, 125);
142 this.btnDelete.Size = new System.Drawing.Size(220, 40);
143 this.btnDelete.Text = "Delete Student";
144 this.btnDelete.Click += new System.EventHandler(this.btnDelete_Click);
145 this.btnDelete.MouseEnter += new System.EventHandler(this.Button_MouseEnter);
146
```

70% No issues found

MAINFORM.cs (MULTIPLE SCREENSHOTS)

The screenshot shows the Microsoft Visual Studio IDE interface. The title bar displays "MainForm.cs" as the active file. The code editor window contains the C# source code for the MainForm class. The code includes several using statements at the top, followed by the namespace declaration. The class itself is annotated with XML documentation comments. The constructor for the class is defined, along with several private member variables and their initializations. The code uses standard C# syntax with color-coded keywords and comments.

```
1  using CIS285Project.Models;
2  using CIS285Project.Services;
3  using System;
4  using System.ComponentModel;
5  using System.Drawing;
6  using System.Linq;
7  using System.Windows.Forms;
8
9  namespace CIS285Project
10 {
11     /// <summary>
12     /// Main user interface for the Student Grade Book application.
13     /// Handles user interaction, validation, and data persistence.
14     /// </summary>
15     public partial class MainForm : Form
16     {
17         private GradeBook _gradeBook = new GradeBook();
18         private BindingList<StudentRecord> _bindingList;
19         private FileStorageService _fileStorage = new FileStorageService();
20         private int _editingIndex = -1;
21
22         private Color buttonNormalColor = Color.FromArgb(0xBF, 0xA7, 0xE2);
23         private Color buttonHoverColor = Color.FromArgb(0xD4, 0xC7, 0xEE);
24
25         public MainForm()
26         {
27             InitializeComponent();
28             InitializeDataBinding();
29             StyleButtons();
30             LoadSampleStudents();
31             UpdateSummary();
32         }
33     }
```



The screenshot shows a Windows application window titled "CIS285Project". The main area displays a table of student records:

Name	Grade 1	Grade 2	Grade 3	Grade 4	Grade 5	Grade 6	Grade 7	Grade 8
Jake Arrieta	85	92	88	91	89	94	90	93
Anthony Rizzo	78	82	80	75	79	81	77	83
Troy Polamalu	94	90	96	98	99	97	95	99
Hines Ward	67	72						

Below the table, there is a status bar with the text "C:\Users\jake\source\repos\CIS285Project\CIS285Project\MainForm.cs 136 lines of code".

```
FileStorageService.cs      GradeBook.cs      MainForm.cs      X StudentRecord.cs      MainForm.Designer.cs
C# CIS285Project          CIS285Project.MainForm      gradeBook
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
```

```
    }
    tb.BackColor = tbOriginalBackColor;
}

// Live validation for grade textboxes
private void GradeTextBox_TextChanged(object sender, EventArgs e)
{
    if (sender is TextBox tb)
    {
        if (string.IsNullOrWhiteSpace(tb.Text))
        {
            tb.BackColor = Color.White;
            return;
        }

        if (double.TryParse(tb.Text, out double value) && value >= 0 && value <= 100)
            tb.BackColor = Color.White;
        else
            tb.BackColor = Color.MistyRose;
    }
}

// -----
// LOAD 19 SAMPLE STUDENTS (8 GRADES)
//

private void LoadSampleStudents()
{
    _gradeBook.Students.Clear();

    _gradeBook.AddStudent(new StudentRecord
    {
        Name = "Jake Arrieta",
        Grade1 = 85,
        Grade2 = 92,
        Grade3 = 88,
        Grade4 = 91,
        Grade5 = 89,
        Grade6 = 94,
        Grade7 = 90,
        Grade8 = 93
    });

    _gradeBook.AddStudent(new StudentRecord
    {
        Name = "Anthony Rizzo",
        Grade1 = 78,
        Grade2 = 82,
        Grade3 = 80,
        Grade4 = 75,
        Grade5 = 79,
        Grade6 = 81,
        Grade7 = 77,
        Grade8 = 83
    });

    _gradeBook.AddStudent(new StudentRecord
    {
        Name = "Troy Polamalu",
        Grade1 = 94,
        Grade2 = 90,
        Grade3 = 96,
        Grade4 = 98,
        Grade5 = 99,
        Grade6 = 97,
        Grade7 = 95,
        Grade8 = 99
    });

    _gradeBook.AddStudent(new StudentRecord
    {
        Name = "Hines Ward",
        Grade1 = 67,
        Grade2 = 72,
    });
}
```

FileStorageService.cs GradeBook.cs **MainForm.cs** StudentRecord.cs MainForm.Designer.cs

CIS285Project CIS285Project.MainForm

```
340      Grade7 = 87,  
341      Grade8 = 100,  
342  
343      };  
344  
345      _bindingList = new BindingList<StudentRecord>(_gradeBook.Students);  
346      dgvStudents.DataSource = _bindingList;  
347  
348      lblStatus.Text = "Loaded 19 active students.";  
349  
350  
351      /// <summary>  
352      /// Updates all class summary statistics displayed in the UI,  
353      /// including averages and passing counts.  
354      /// </summary>  
355      private void UpdateSummary()  
356      {  
357          int count = _gradeBook.Students.Count;  
358          lblTotalStudents.Text = $"Total Students: {count}";  
359          lblSummaryClassAvg.Text = $"Class Avg: {_gradeBook.CalculateClassAverage():F2}";  
360  
361          if (count > 0)  
362          {  
363              lblHighestAvg.Text = $"Highest Avg: {_gradeBook.Students.Max(s => s.Average):F2}";  
364              lblLowestAvg.Text = $"Lowest Avg: {_gradeBook.Students.Min(s => s.Average):F2}";  
365              lblPassingCount.Text = $"Passing (>=70): {_gradeBook.Students.Count(s => s.Average >= 70)}";  
366          }  
367          else  
368          {  
369              lblHighestAvg.Text = "Highest Avg: N/A";  
370              lblLowestAvg.Text = "Lowest Avg: N/A";  
371              lblPassingCount.Text = "Passing: 0";  
372          }  
373  
374  
375      // -----  
376      // BUTTON CLICK EVENTS  
377      // -----  
378  
379      private void btnAdd_Click(object sender, EventArgs e)  
380      {  
381          ClearInputs();  
382          _editingIndex = -1;  
383          lblStatus.Text = "Adding new student...";  
384      }  
385  
386      private void btnEdit_Click(object sender, EventArgs e)  
387      {  
388          if (dgvStudents.CurrentRow == null)  
389          {  
390              MessageBox.Show("Select a student to edit.");  
391              return;  
392          }  
393  
394          _editingIndex = dgvStudents.CurrentRow.Index;  
395          var s = _gradeBook.Students[_editingIndex];  
396  
397          txtName.Text = s.Name;  
398          txtGrade1.Text = s.Grade1?.ToString();  
399          txtGrade2.Text = s.Grade2?.ToString();  
400          txtGrade3.Text = s.Grade3?.ToString();  
401          txtGrade4.Text = s.Grade4?.ToString();  
402          txtGrade5.Text = s.Grade5?.ToString();  
403          txtGrade6.Text = s.Grade6?.ToString();  
404          txtGrade7.Text = s.Grade7?.ToString();  
405          txtGrade8.Text = s.Grade8?.ToString();  
406          txtNotes.Text = s.Notes;  
407  
408          lblStatus.Text = "Editing student...";  
409      }
```

70% ▾ 0 ▲ 1 ↑ ↓ | ⌂ ▾ Ln: 13 Ch: 53

FileStorageService.cs | GradeBook.cs | StudentRecord.cs | **MainForm.cs** | Main.cs

CIS285Project

```
529     }
530 
531     1 reference
532     private void btnAverage_Click(object sender, EventArgs e)
533     {
534         lblClassAverage.Text = _gradeBook.CalculateClassAverage().ToString("F2");
535         lblStatus.Text = "Class average updated.";
536         UpdateSummary();
537     }
538 
539     1 reference
540     private void btnSaveFile_Click(object sender, EventArgs e)
541     {
542         SaveFileDialog dialog = new SaveFileDialog();
543         dialog.Filter = "XML Files|*.xml";
544 
545         if (dialog.ShowDialog() == DialogResult.OK)
546         {
547             _fileStorage.Save(dialog.FileName, _gradeBook.Students);
548             lblStatus.Text = "File saved.";
549         }
550 
551     1 reference
552     private void btnSaveNotes_Click(object sender, EventArgs e)
553     {
554         if (dgvStudents.CurrentRow == null)
555         {
556             MessageBox.Show("Select a student first.");
557             return;
558         }
559 
560         int index = dgvStudents.CurrentRow.Index;
561         _gradeBook.Students[index].Notes = txtNotes.Text;
562 
563         lblStatus.Text = "Notes saved.";
564 
565     1 reference
566     private void btnLoadFile_Click(object sender, EventArgs e)
567     {
568         OpenFileDialog dialog = new OpenFileDialog();
569         dialog.Filter = "XML Files|*.xml";
570 
571         if (dialog.ShowDialog() == DialogResult.OK)
572         {
573             var list = _fileStorage.Load(dialog.FileName);
574 
575             _gradeBook.Students.Clear();
576             _gradeBook.Students.AddRange(list);
577 
578             _bindingList = new BindingList<StudentRecord>(_gradeBook.Students);
579             dgvStudents.DataSource = _bindingList;
580 
581             UpdateSummary();
582             lblStatus.Text = "File loaded.";
583         }
584     }
585 }
```

PROGRAM.cs

The screenshot shows a code editor interface with a dark theme. On the left, there's a vertical toolbar labeled "Data Sources". The main window displays the "Program.cs" file under the "CIS285Project" solution. The code is as follows:

```
1  using System;
2  using System.Windows.Forms;
3
4  namespace CIS285Project
5  {
6      internal static class Program
7      {
8          [STAThread]
9          static void Main()
10         {
11             Application.EnableVisualStyles();
12             Application.SetCompatibleTextRenderingDefault(false);
13             Application.Run(new MainForm()); // YOUR MAIN WINDOW
14         }
15     }
16 }
17
```

FILESTORAGESERVICE.cs

```
10  namespace CIS285Project.Services
11  {
12      /// <summary>
13      /// Provides functionality for saving and loading student records
14      /// to and from XML files.
15      /// </summary>
16      2 references
17      public class FileStorageService
18      {
19          /// <summary>
20          /// Saves a list of student records to an XML file.
21          /// </summary>
22          /// <param name="filePath">The file path where the XML file will be saved.</param>
23          /// <param name="students">The list of student records to save.</param>
24          1 reference
25          public void Save(string filePath, List<StudentRecord> students)
26          {
27              var doc = new XDocument(
28                  new XElement("Students",
29                      students.Select(s =>
30                          new XElement("Student",
31                              new XAttribute("Name", s.Name ?? ""),
32                              new XAttribute("Grade1", s.Grade1?.ToString() ?? ""),
33                              new XAttribute("Grade2", s.Grade2?.ToString() ?? ""),
34                              new XAttribute("Grade3", s.Grade3?.ToString() ?? ""),
35                              new XAttribute("Grade4", s.Grade4?.ToString() ?? ""),
36                              new XAttribute("Grade5", s.Grade5?.ToString() ?? ""),
37                              new XAttribute("Grade6", s.Grade6?.ToString() ?? ""),
38                              new XAttribute("Grade7", s.Grade7?.ToString() ?? ""),
39                              new XAttribute("Grade8", s.Grade8?.ToString() ?? ""),
40                              new XElement("Notes", s.Notes ?? ""))
41                      )
42                  )
43              );
44
45              doc.Save(filePath);
46          }
47
48          /// <summary>
49          /// Loads student records from an XML file.
50          /// </summary>
51          /// <param name="filePath">The file path of the XML file to load.</param>
52          /// <returns>A list of student records loaded from the file.</returns>
53          1 reference
54          public List<StudentRecord> Load(string filePath)
55          {
56              var result = new List<StudentRecord>();
57
58              if (!File.Exists(filePath))
59                  return result;
60
61              var doc = XDocument.Load(filePath);
62              foreach (var elem in doc.Root.Elements("Student"))
63              {
64                  var s = new StudentRecord
65                  {
66                      Name = (string)elem.Attribute("Name")
67                  };
68
69                  s.Grade1 = ParseNullableDouble((string)elem.Attribute("Grade1"));
70                  s.Grade2 = ParseNullableDouble((string)elem.Attribute("Grade2"));
71                  s.Grade3 = ParseNullableDouble((string)elem.Attribute("Grade3"));
72                  s.Grade4 = ParseNullableDouble((string)elem.Attribute("Grade4"));
73                  s.Grade5 = ParseNullableDouble((string)elem.Attribute("Grade5"));
74                  s.Grade6 = ParseNullableDouble((string)elem.Attribute("Grade6"));
75                  s.Grade7 = ParseNullableDouble((string)elem.Attribute("Grade7"));
76                  s.Grade8 = ParseNullableDouble((string)elem.Attribute("Grade8"));
77
78                  s.Notes = (string)elem.Element("Notes") ?? "";
79
80                  result.Add(s);
81              }
82          }
83      }
84  }
```

GRADEBOOK.cs

The screenshot shows the Microsoft Visual Studio IDE interface with the following details:

- Solution Explorer:** On the left, it shows the project structure with "CIS285Project" as the root.
- Code Editor:** The main window displays the `GradeBook.cs` file under the `CIS285Project` namespace.
- Toolbars:** Standard Visual Studio toolbars for file operations (File, Edit, View, Tools, Help) are visible at the top.
- Status Bar:** The bottom status bar shows the path as "C:\Users\...".

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace CIS285Project.Models
8  {
9      /// <summary>
10     /// Manages a collection of student records and provides
11     /// methods for modifying students and calculating class statistics.
12     /// </summary>
13     public class GradeBook
14     {
15         /// <summary>
16         /// Gets the list of students in the grade book.
17         /// </summary>
18         public List<StudentRecord> Students { get; private set; }
19
20         /// <summary>
21         /// Initializes a new instance of the <see cref="GradeBook"/> class.
22         /// </summary>
23         public GradeBook()
24         {
25             Students = new List<StudentRecord>();
26         }
27
28         /// <summary>
29         /// Adds a new student to the grade book.
30         /// </summary>
31         /// <param name="s">The student record to add.</param>
32         public void AddStudent(StudentRecord s) => Students.Add(s);
33
34         /// <summary>
35         /// Updates an existing student at the specified index.
36         /// </summary>
37         /// <param name="index">The index of the student to update.</param>
38         /// <param name="s">The updated student record.</param>
39         public void UpdateStudent(int index, StudentRecord s)
40         {
41             if (index >= 0 && index < Students.Count)
42                 Students[index] = s;
43         }
44
45         /// <summary>
46         /// Removes a student from the grade book at the specified index.
47         /// </summary>
48         /// <param name="index">The index of the student to remove.</param>
49         public void RemoveStudent(int index)
50         {
51             if (index >= 0 && index < Students.Count)
52                 Students.RemoveAt(index);
53         }
54
55         /// <summary>
56         /// Calculates the average grade across all students.
57         /// </summary>
58         /// <returns>The class average.</returns>
59         public double CalculateClassAverage()
60         {
61             if (Students.Count == 0)
62                 return 0;
63
64             return Students.Average(s => s.Average);
65         }
66     }
67 }
```

STUDENTRECORD.cs

The screenshot shows a Windows application window with multiple tabs open in a code editor. The active tab is 'StudentRecord.cs'. The code in the tab is as follows:

```
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace CIS285Project.Models
8  {
9      /// <summary>
10     /// Represents a student record containing the student's name,
11     /// grades, notes, and calculated average.
12     /// </summary>
13     public class StudentRecord
14     {
15         /// <summary>
16         /// The student's full name.
17         /// </summary>
18         public string Name { get; set; }
19
20         /// <summary>First grade.</summary>
21         public double? Grade1 { get; set; }
22
23         /// <summary>Second grade.</summary>
24         public double? Grade2 { get; set; }
25
26         /// <summary>Third grade.</summary>
27         public double? Grade3 { get; set; }
28
29         /// <summary>Fourth grade.</summary>
30         public double? Grade4 { get; set; }
31
32         /// <summary>Fifth grade.</summary>
33         public double? Grade5 { get; set; }
34
35         /// <summary>Sixth grade.</summary>
36         public double? Grade6 { get; set; }
37
38         /// <summary>Seventh grade.</summary>
39         public double? Grade7 { get; set; }
40
41         /// <summary>Eighth grade.</summary>
42         public double? Grade8 { get; set; }
43
44         /// <summary>
45         /// Optional notes associated with the student.
46         /// </summary>
47         public string Notes { get; set; }
48
49
50         /// <summary>
51         /// Calculates the student's average grade based on all
52         /// non-null grade values.
53         /// </summary>
54         public double Average
55         {
56             get
57             {
58                 var grades = new double[]
59                 {
60                     Grade1, Grade2, Grade3, Grade4,
61                     Grade5, Grade6, Grade7, Grade8
62                 };
63
64                 var valid = grades.Where(g => g.HasValue).Select(g => g.Value);
65             }
66         }
67     }
```

Source Code Reference:

This project was developed by our group and was not adapted from any external repository or tutorial project.