

### **Student Course Collection**

Create an application that collects information about students and the courses they have completed, and holds this information in collection, implemented as a List or ArrayList object.

- Create a class named Course with the following properties: CourseId As String, Credits As Integer, Grade As Double.
- Add a class-level variable to the Student class that holds a list of courses.
- Create a new form named *Course Information* that contains the following input fields: course ID, credits, and grade. The list box displays all courses in the student's transcript.
- On the main form, add a *Course Information* button. Use this button to display the Course Information form. The button should initially be disabled; it is enabled when the user clicks either the *Add to Collection* button or the *Find by ID* button.

### **Input Requirements**

The following input requirements are designed to prevent the program from throwing an exception or storing invalid data:

1. No two Student IDs can be the same.
2. No input fields can be blank.
3. The Grade field must be a positive numeric value between 0.0 and 4.0.
4. The course credits field must be a positive integer between 0 and 6.
5. A duplicate course number cannot be added to the list of courses.
6. Error messages must be specific, identifying exactly which field has a missing or incorrect value.

Each of these requirements is included in the testing steps that we have outlined for this programming challenge.

### Suggestions

1. In the main form, create a class-level variable of type Student so it can be accessible to different event handlers. This will affect the event handlers for both the *Add* and *Find* buttons.
2. The Student class should have a ReadOnly property that exposes its ArrayList. This will be useful when your program needs to display the courses in a list box or add a new course to the list.
3. The Form\_Load event handler for the Course Information form should fill the list box with the list of courses belonging to the current student. It should also clear all text boxes on the form.
4. Use the ErrorProvider control to display error messages.

The screenshot shows a Windows form titled "Student Course Collection". It features three text input fields: "Student ID:" with the value "20000", "Last name:" with the value "Jones", and "Test average:" with the value "80". To the right of these fields are two buttons: "Add to Collection" and "Course Information". Below the input fields, there is a "View All" button (highlighted with a blue border) and a "Find by ID" button. Under "Find by ID" is a small empty text box. To the right of these buttons is a list box containing two entries: "10000, Smith, Test average = 87.50" and "20000, Jones, Test average = 80.00".

The screenshot shows a Windows form titled "Course Information Form". At the top, it displays "Student name: 20000, Jones". Below this is a section titled "Add Course Info" which contains three text input fields: "Course number" with the value "CS101", "Credits" with the value "5", and "Grade" with the value "3.5". To the right of these fields is a "Save" button (highlighted with a blue border). Below the "Add Course Info" section is a "Close" button. To the right of the "Add Course Info" section is a list box titled "Existing Courses:" which contains one entry: "CS101, 5 CR, grade: 3.5".