

Assign 06

Due date and time: 04/25/2023 11:59 pm

Total points: 20

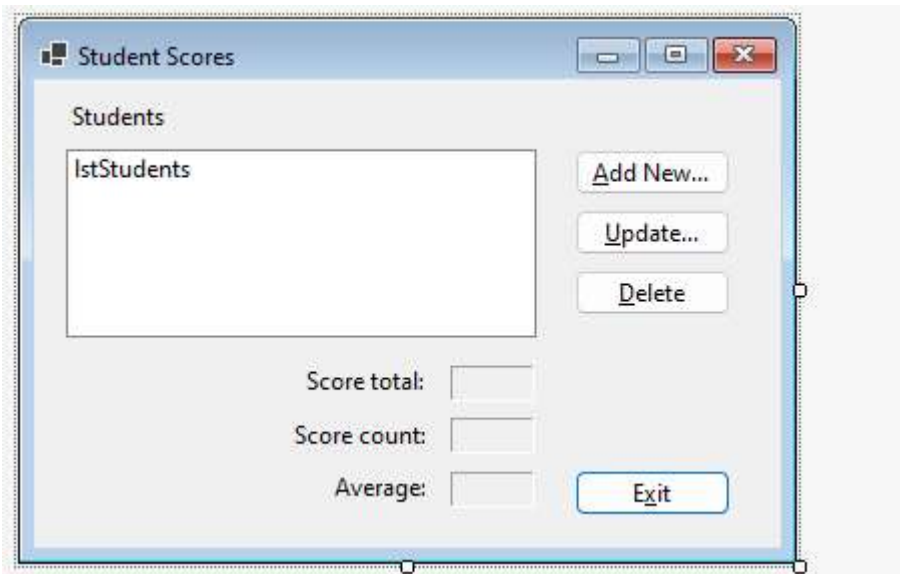
Description: In this assignment, you will create a multiform desktop application that let the user maintain a list of student scores. You will use a text file to store the student data.

Specifications

- When this application starts, it should read student data from a text file named StudentScores.txt in the C:\C#\Files directory and display this data in the Students list box. If this directory or file doesn't exist, it should be created.
- When students are added, updated, or deleted, this application should write the student data to the StudentScores.txt file, overwriting any existing data.
- This application should use a class named StudentDB to read data from and write data to the StudentScores.txt file.
- The students should be displayed in alphabetical order in the list box.
- This application should use a class named Student to store information about each student.
- The Student class should override the ToString() method of the Object class to provide a string representation of each Student object that includes the student name and scores as shown above.
- The Update Student Scores form should create a clone of the current student object and then apply changes to the clone. That way, the changes will be saved to the current student only if the user clicks the OK button. To create a clone, the Student class will need to implement the ICloneable interface, and the Clone() method will need to implement a deep copy.
- This application should make sure that the user enters a name for a new student. However, a new student can be added without any scores.
- This application should check all scores entered by the user to make sure that each score is a valid integer from 0 to 100.
- Add data validation.
 - Required field, numeric, range 0-100.

- Create a separate Validator.cs static class to write data validation code.
- Add exception handling.

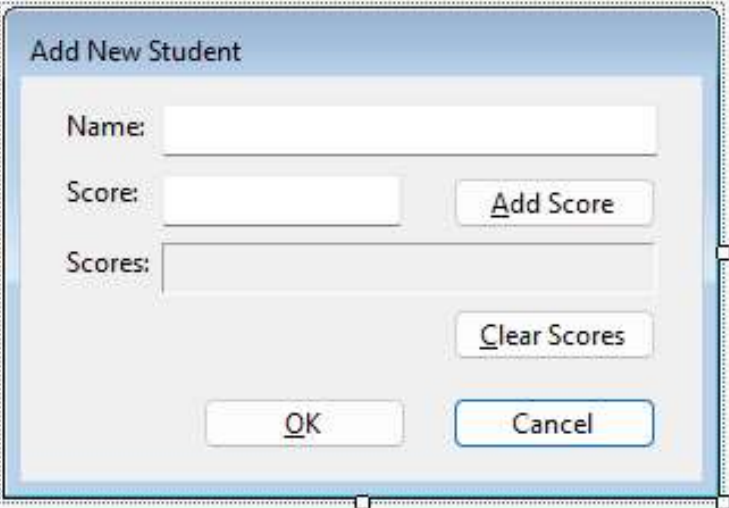
The Student Scores form



Operation:

- To display the total, count, and average for a student, the user selects the student from the list box. If the list box is empty, the total, count, and average labels should be cleared.
- To add a new student, the user clicks the Add New button to display the Add New Student dialog box.
- To update an existing student's scores, the user selects the student in the list box and clicks the Update button to display the Update Student Scores dialog box.
- To delete a student, the user selects the student in the list box and clicks the Delete button.

Add New student form.



The image shows a Windows-style dialog box titled "Add New Student". It contains three input fields: "Name:", "Score:", and "Scores:". The "Score:" field has an "Add Score" button to its right. The "Scores:" field has a "Clear Scores" button below it. At the bottom of the dialog are "OK" and "Cancel" buttons. The dialog has a blue title bar and a light gray body.

Operation

- To add a new student, the user enters a student name and, optionally, one or more scores and clicks the OK button.
- To add a score, the user enters a score and clicks the Add Score button. The score is added to the list of scores in the Scores label.
- To remove all scores from the Scores label, the user clicks the Clear Scores button.
- To cancel the add operation, the user clicks the Cancel button.

The Update Student Scores and Add/Update Score forms.

After clicking Update on Student Scores form, Click Add on Update Student Scores form, then add a score on Add Score.

The image displays three screenshots of a user interface for managing student scores.

Update Student Scores: This dialog box has a title bar "Update Student Scores". It contains a "Name:" label followed by a text input field with the value "Joel Murach". Below this is a "Scores:" label followed by a list box containing the values 97, 91, and 83. The value 97 is currently selected. To the right of the list box are four buttons: "Add", "Update", "Remove", and "Clear Scores". At the bottom of the dialog are two buttons: "OK" and "Cancel".

Add Score: This dialog box has a title bar "Add Score". It contains a "Score:" label followed by a text input field with the value 88. Below the input field are two buttons: "Add" and "Cancel".

Update Score: This dialog box has a title bar "Update Score". It contains a "Score:" label followed by a text input field with the value 97. Below the input field are two buttons: "Update" and "Cancel".

Operation

- To add a score, the user clicks the Add button and enters the score in the Add Score dialog box that's displayed.

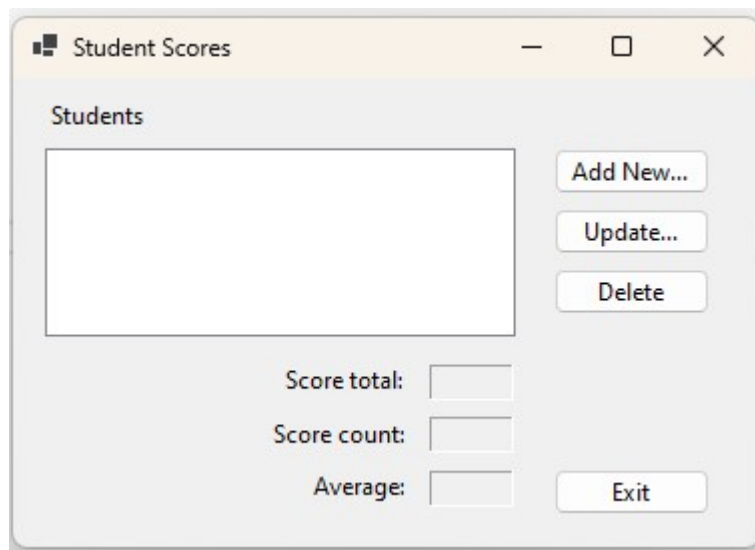
This screenshot shows the "Update Score" dialog box. It features a title bar "Update Score". The main area contains a "Score:" label and a text input field with the value "97". At the bottom, there are two buttons: "Update" and "Cancel".

- To update a score, the user selects the score, clicks the Update button, and changes the score in the Update Score dialog box that's displayed.
- To remove a score from the Scores list box, the user selects the score and clicks the Remove button.
- To remove all scores from the Scores list box, the user clicks the Clear Scores button.
- To accept all changes, the user clicks the OK button.
- To cancel the update operation, the user clicks the Cancel button.

Hint

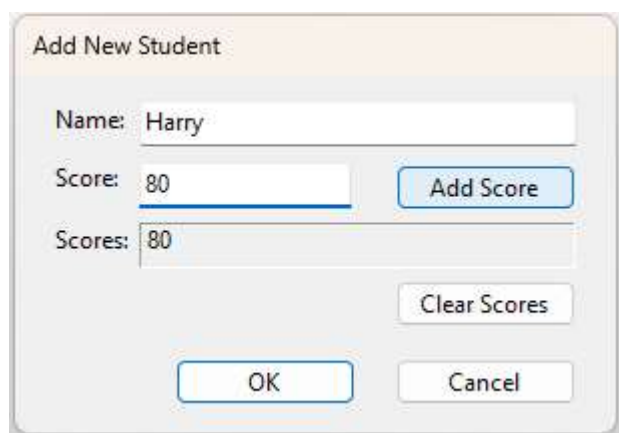
- To get the labels that display the score total, score count, and average on the Student Scores form, the scores on the Add New Student form, and the name on the Update Student Scores form to look like the ones shown above, you'll need to set the AutoSize property of the labels to False and the BorderStyle property to Fixed3D.

When you run this application – happy scenario.



The 'Student Scores' window features a title bar with standard Windows controls. The main area is titled 'Students' and contains a large empty rectangular box. To the right of this box are three buttons: 'Add New...', 'Update...', and 'Delete'. Below the box, there are three labels with corresponding input fields: 'Score total:', 'Score count:', and 'Average:'. An 'Exit' button is located at the bottom right of the window.

Click Add New.



The 'Add New Student' dialog box has a title bar. It contains a 'Name:' label followed by a text box with 'Harry' entered. Below this is a 'Score:' label followed by a text box with '80' entered, and an 'Add Score' button to its right. Underneath the 'Score' section is a 'Scores:' label followed by a text box with '80' entered. To the right of the 'Scores' text box is a 'Clear Scores' button. At the bottom of the dialog are 'OK' and 'Cancel' buttons.

The 'Student Scores' window displays a list of students. 'Harry' is selected with a score of 80. The summary statistics show a total score of 80, a count of 1, and an average of 80.

Students
Harry 80

Score total: 80
Score count: 1
Average: 80

Buttons: Add New..., Update..., Delete, Exit

Add one more student.

The 'Add New Student' dialog box allows adding a new student. 'Shallu' is entered as the name, and '90' is entered as the score. The 'Add Score' button is visible.

Name: Shallu
Score: 90
Scores: 90

Buttons: Add Score, Clear Scores, OK, Cancel

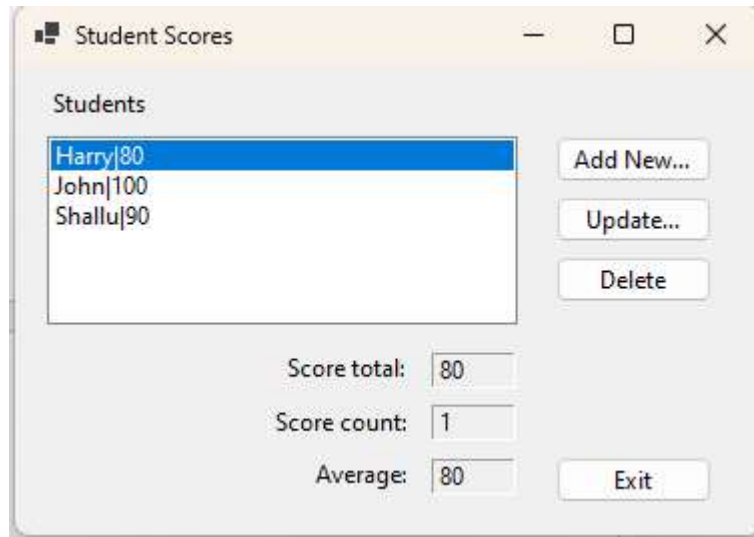
The 'Student Scores' window now shows two students: 'Harry' with a score of 80 and 'Shallu' with a score of 90. The summary statistics remain the same as before.

Students
Harry 80
Shallu 90

Score total: 80
Score count: 1
Average: 80

Buttons: Add New..., Update..., Delete, Exit

After adding John – pay attention to the sorting of names.

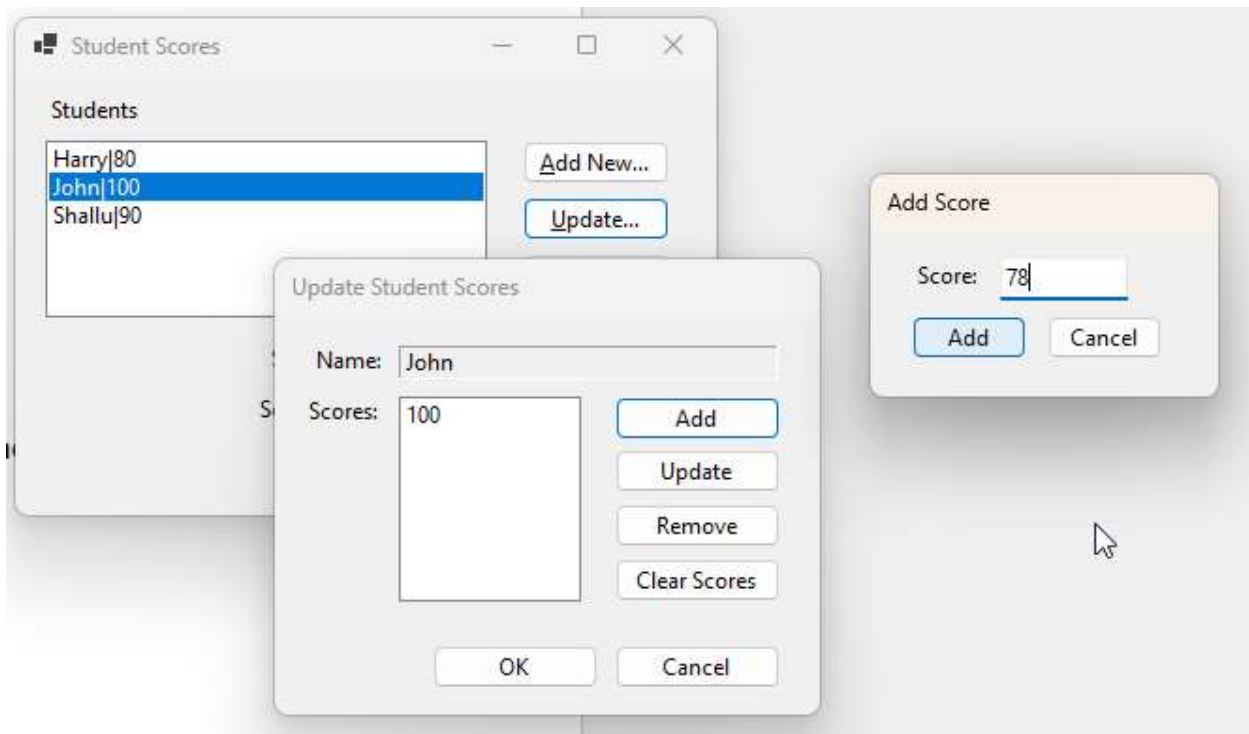


The 'Student Scores' window displays a list of students in a table. The first student, Harry, is selected. To the right of the table are buttons for 'Add New...', 'Update...', and 'Delete'. Below the table, there are three summary fields: 'Score total: 80', 'Score count: 1', and 'Average: 80'. An 'Exit' button is located at the bottom right.

Students
Harry 80
John 100
Shallu 90

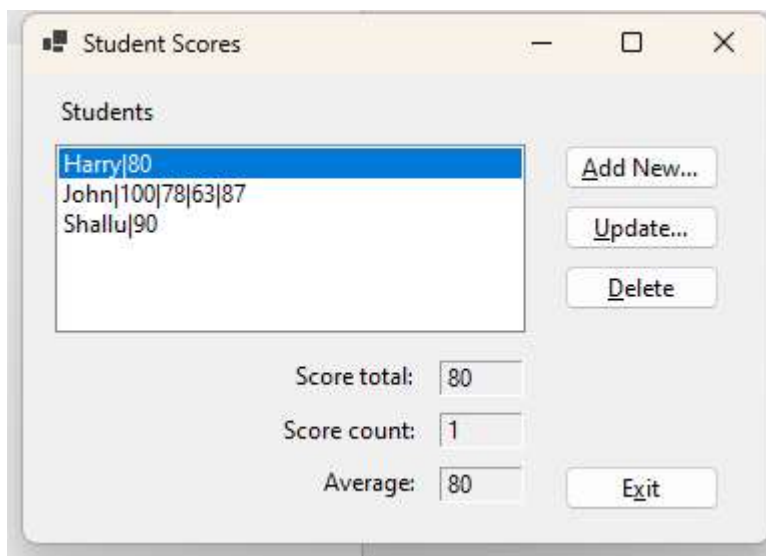
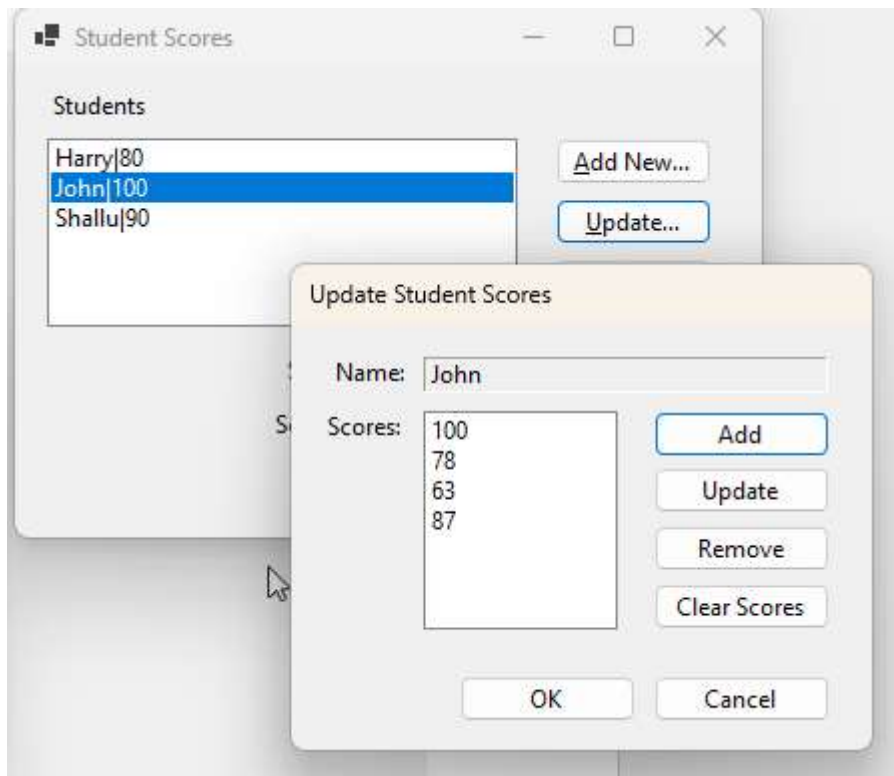
Score total: 80
Score count: 1
Average: 80

Let's update and add one more score to John

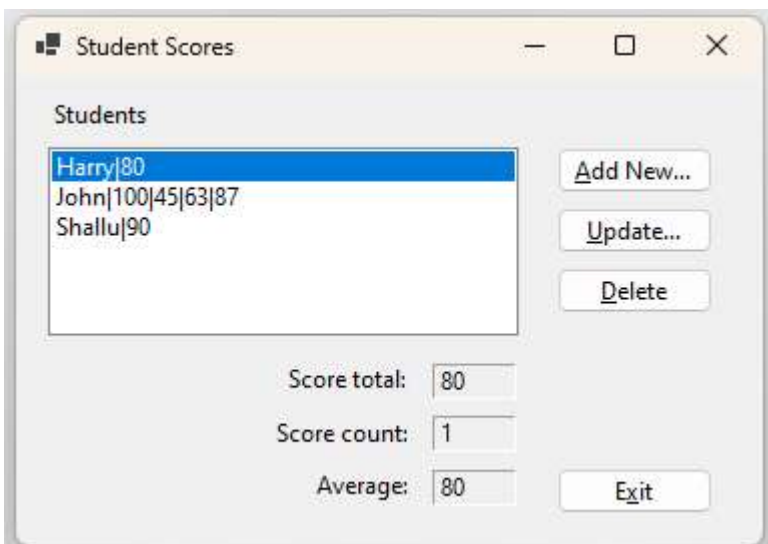
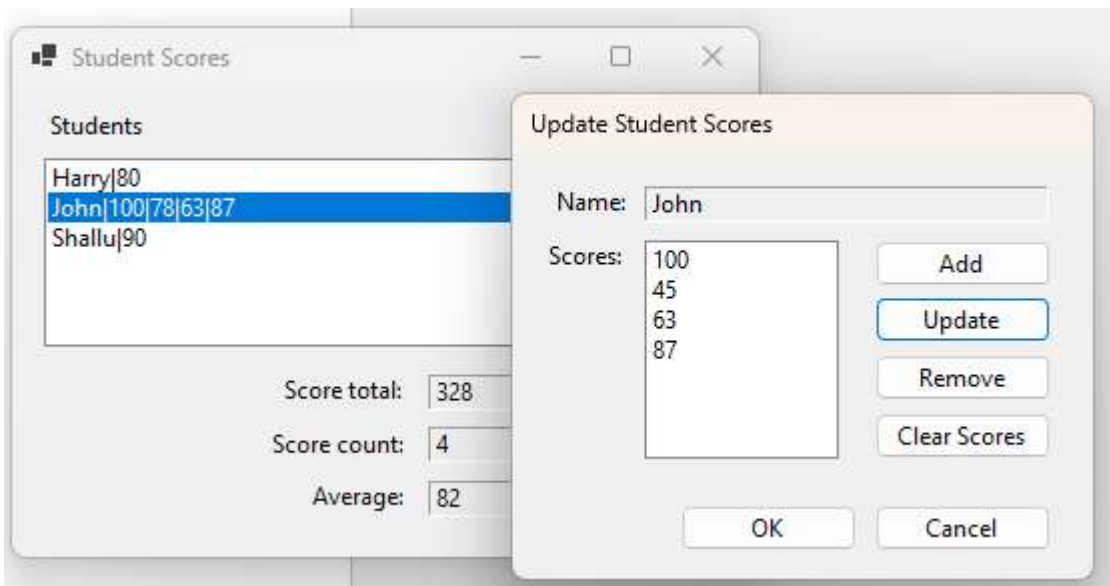
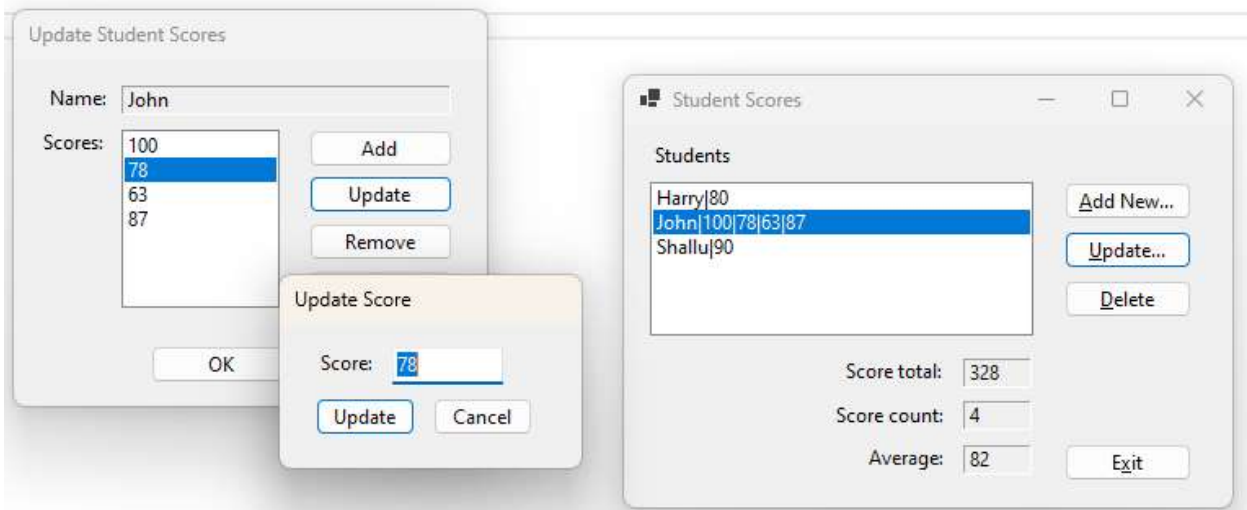


The 'Student Scores' window is shown with 'John|100' selected. An 'Update Student Scores' dialog is open, showing 'Name: John' and 'Scores: 100'. It has buttons for 'Add', 'Update', 'Remove', 'Clear Scores', 'OK', and 'Cancel'. To the right, an 'Add Score' dialog is open, showing 'Score: 78' and buttons for 'Add' and 'Cancel'.

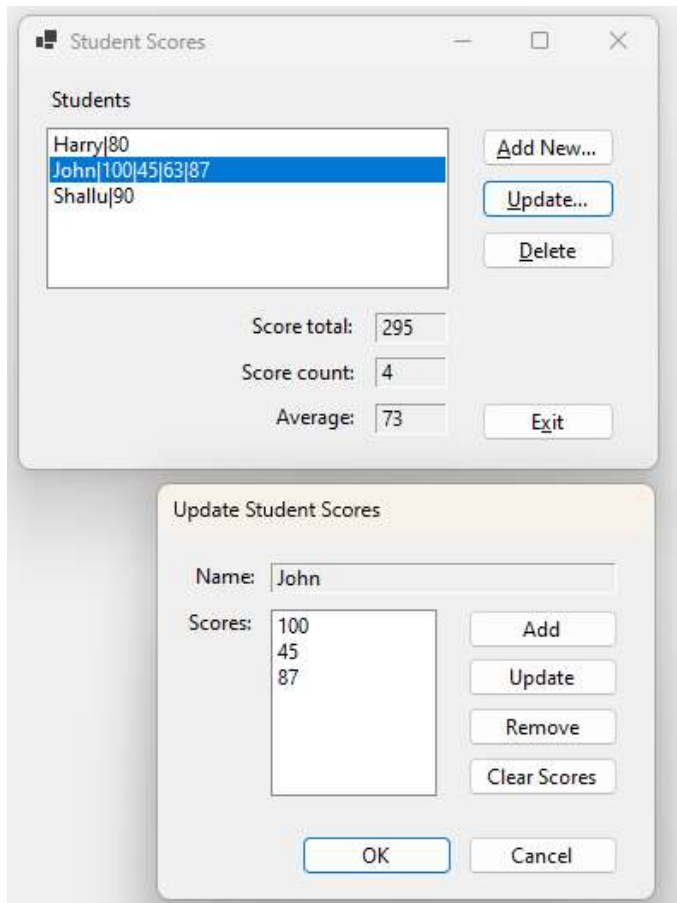
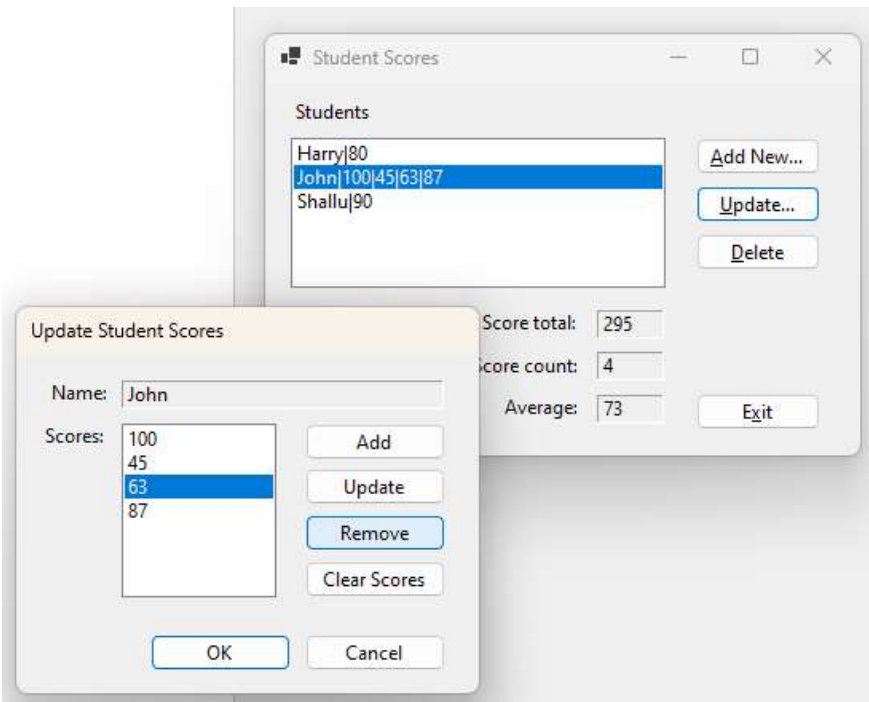
After adding a few more scores.



Let's update score 78 to 45 for Student John.



Let's remove 63 score for John.



Delete Student Shallu

Student Scores

Students

Harry 80
John 100 45 87

Add New...
Update...
Delete

Score total: 80
Score count: 1
Average: 80

Exit

Let's clear all scores for John.

Student Scores

Students

Harry 80
John 100 45 87

Add New...
Update...
Delete

Score total: 232
Score count: 3
Average: 77

Exit

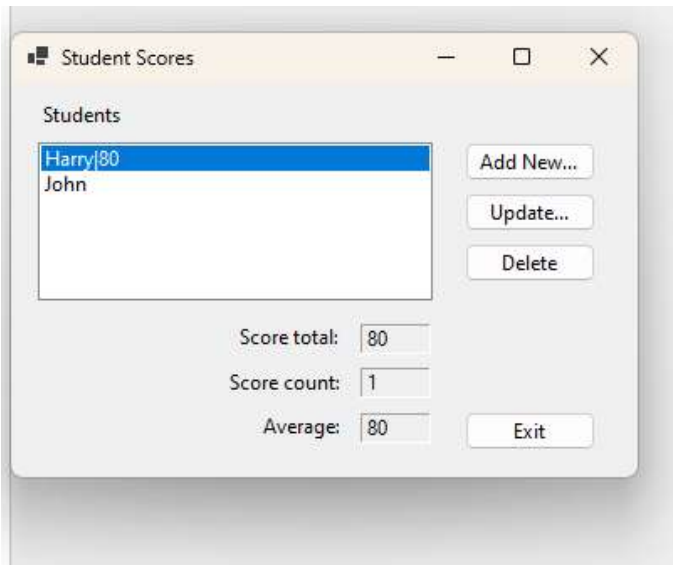
Update Student Scores

Name: John

Scores: 100, 45, 87

Add
Update
Remove
Clear Scores

OK Cancel

**Deliverable:**

1. C# multiform desktop application.

Submission:

1. A compressed zip folder of C# solution to canvas Assign 06 folder.
2. In a separate Word document, include a retrospective that discusses:
 - a. What went well with the assignment?
 - b. What did not go well -- what did you struggle with?

Note:

1. The assignment awards 20 points according to the criteria given above. Partial or incorrect completion of the elements will reduce points awarded.
2. The retrospective will not be graded but it is required. Failure to include a retrospective will result in a 5% reduction in points.
3. Use best practices to write code as discussed during class lectures.