Previous Unit 2 of 7 Next

✓ 100 XP

# Prepare

9 minutes

In this guided project, you'll use Visual Studio Code to develop a C# application. The application will use arrays, `foreach` statements, and `if` statements to implement a list of design parameters. You'll begin by creating the array variables that contain your application data. To complete the project, you'll develop `foreach` and `if` statements that implement the application design goals.

> ⓘ **Important**
>
> This module includes coding activities that require Visual Studio Code. You'll need access to a development environment that has Visual Studio Code installed and configured for C# application development.

## Project overview

You're developing a Student Grading application that automates the calculation of grades for each student in a class. The parameters for your application are:

- Create a C# console application.
- Start with four students. Each student has five exam scores.
- Each exam score is an integer value, 0-100, where 100 represents 100% correct.
- A student's overall exam score is the average of their five exam scores.
- Criteria for extra credit assignments:
  - Include extra credit assignment scores in the student's scores array.
  - Extra credit assignments are worth 10% of an exam score (when calculating the final numeric grade).
  - Add extra credit assignment scores to the student's total exam score before calculating the final numeric grade.

- Your application needs to automatically assign letter grades based on the calculated final score for each student.
- Your application needs to output/display each student’s name and formatted grade.
- Your application needs to support adding other students and scores with minimal impact to the code.

You've already completed an initial version of the application. The Starter code project for this Guided project module includes a Program.cs file that provides the following code features:

- The code declares variables used to define student names and individual exam scores for each student.
- The code includes the variables and algorithms used to sum the exam scores and calculate the average exam score for each student.
- The code includes a hard coded letter grade that the developer must apply manually.
- The code includes Console.WriteLine() statements to display the student grading report.

Your goal is to update the existing code to include the following features:

- Use arrays to store student names and assignment scores.
- Use a `foreach` statement to iterate through the student names as an outer program loop.
- Use an `if` statement within the outer loop to identify the current student name and access that student's assignment scores.
- Use a `foreach` statement within the outer loop to iterate through the assignment scores array and sum the values.
- Use an updated algorithm within the outer loop to calculate the average exam score for each student.
- Use an `if-elseif-else` construct within the outer loop to evaluate the average exam score and assign a letter grade automatically.
- Integrate extra credit scores when calculating the student's final score and letter grade as follows:
  - Your code must detect extra credit assignments based on the number of elements in the student's scores array.
  - Your code must apply the 10% weighting factor to extra credit assignments before adding extra credit scores to the sum of exam scores.

The following list shows the letter grade that corresponds to numeric scores:

Output

```
97 - 100   A+
93 - 96    A
90 - 92    A-
87 - 89    B+
83 - 86    B
80 - 82    B-
77 - 79    C+
73 - 76    C
70 - 72    C-
67 - 69    D+
63 - 66    D
60 - 62    D-
0  - 59    F
```

The update application needs to produce a formatted student grading report that appears as follows:

Output

```
Student         Grade

Sophia:         92.2    A-
Andrew:         89.6    B+
Emma:           85.6    B
Logan:          91.2    A-
```

# Setup

Use the following steps to prepare for the Guided project exercises.

1. To download a zip file containing the Starter project code, select the following link: Lab Files.

2. Unzip the download files.

   Unzip the files in your development environment. Consider using your PC as your development environment so that you have access to your code after completing this module. If you aren't using your PC as your development environment, you can unzip the files in a sandbox or hosted environment.

  a. On your local machine, navigate to your downloads folder.
  b. Right-click **Guided-project-foreach-if-array-CSharp.main.zip**, and then select **Extract all**.
  c. Select **Show extracted files when complete**, and then select **Extract**.
  d. Make note of the extracted folder location.

3. Copy the extracted **GuidedProject** folder to your Windows Desktop folder.

   > **Note**
   >
   > If a folder named **GuidedProject** already exists, you can select **Replace the files in the destination** to complete the copy operation.

4. Open the new **GuidedProject** folder in Visual Studio Code.
   a. Open Visual Studio Code in your development environment.
   b. In Visual Studio Code, on the **File** menu, select **Open Folder**.
   c. Navigate to the Windows Desktop folder and locate the "GuidedProject" folder.
   d. Select **GuidedProject** and then select **Select Folder**.

   The Visual Studio Code EXPLORER view should show the GuidedProject folder and two subfolders named Final and Starter.

You're now ready to begin the Guided project exercises. Good luck!

## Next unit: Exercise - Create arrays and foreach loops

Continue >