

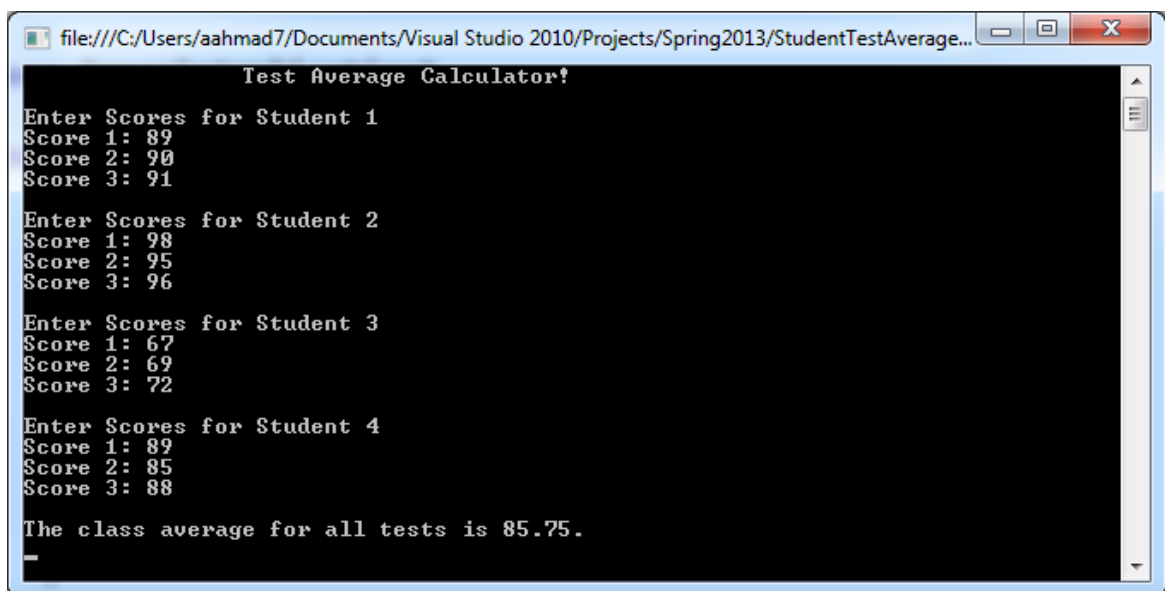
**CIS 345 – Business Information Systems Development II – Fall 2014**

In-class Exercise 4: Two Dimensional Arrays

Due on Blackboard: Today, Monday, September 22 by 10:00 PM

Design a C# program that accepts 3 test scores for a set of 4 students and calculates its average.  
Utilize a 2-dimensional array structure to store student scores.

Student 1: Scores: 89 90 91  
Student 2: Scores: 98 95 96  
Student 3: Scores: 67 69 72  
Student 4: Scores: 89 85 88



```
file:///C:/Users/aahmad7/Documents/Visual Studio 2010/Projects/Spring2013/StudentTestAverage...
Test Average Calculator!
Enter Scores for Student 1
Score 1: 89
Score 2: 90
Score 3: 91

Enter Scores for Student 2
Score 1: 98
Score 2: 95
Score 3: 96

Enter Scores for Student 3
Score 1: 67
Score 2: 69
Score 3: 72

Enter Scores for Student 4
Score 1: 89
Score 2: 85
Score 3: 88

The class average for all tests is 85.75.
```

- Create a project called AverageCalculatorApp and rename Program.cs to AverageCalculator.cs.
- In your Class, AverageCalculator:

Declare a ***static*** class field: a 2-dimensional array of integers called scoreArray and instantiate or allocate the array with 4 rows and 3 columns. Use the int[row, col] structure.

***Implement the following methods within your class. Methods will be static.***

- *Method Name:* PopulateTestScores  
*Purpose:* To prompt the user and read 3 sets of test scores for 4 students.  
*Access Modifier:* Private  
*Parameters:* None  
*Return value:* None

General method logic:

Loop through all the rows (use i as counter variable)

    Tell the user to enter scores for Student #x (use counter variable)

    Loop through all the columns (use j as counter variable)

        Ask the user to enter a score

        Read the score and assign it to the current array element i.e. scoreArray[row counter, column counter]

    End loop

End loop

- *Method Name:* CalculateTestAverage  
*Purpose:* To loop through all the scores and calculate an overall average  
*Access Modifier:* Private  
*Parameters:* None  
*Return value:* None

General method logic:

Declare an integer called total and initialize it to 0.  
Declare a double called average and initialize it to 0.0.

Loop through all the rows (use i as counter variable)

    Loop through all the columns (use j as counter variable)

        Add the current array element value (use scoreArray[row,col]) to the total and assign that to the total. (*similar logic to total = total + value*).

    End loop

End loop

Calculate average by dividing the total by the array length (cast total as a double).  
Print the average to the Console (use "{0:F2}" for two decimal places).

- *Method Name:* Main  
*Purpose:* To start the program and call other methods.

General method logic:

Call PopulateTestScores

Call CalculateTestAverage

Put in a ReadLine to make the program stop in debug mode.

## Optional Challenges and Enhancements

- Write a method to calculate the average only for Student 1.
  - For added sophistication, take a student index number as a parameter.
- Write a method to calculate the class average for Test 3 (all students but only Test 3).
- Don't ask users to enter scores for users one by one. Instead, ask for student scores separated by a space, e.g. Student 1's scores are input as "89 90 91" in one line and student 2's scores are input as "98 95 96" in one line.
  - Split that string using the Split() method. Split returns you an array of strings. Process that array of strings and convert them into integers for use in your array.
  - Doing this will entail modification in your program structure for user input (PopulateTestScores) but once your array of integers is populated, no more modifications will be needed.

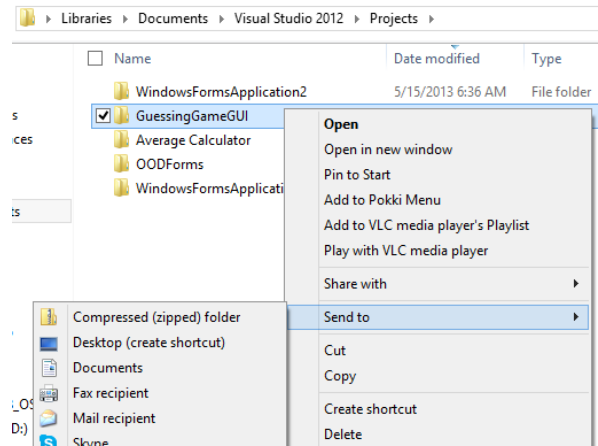
## Learning Exercises

- Create a local array and store in it a reference to the static array.
  - Or have multiple array variables store references to the same array.
- Modify the array using both the references i.e. variable names of the arrays.
  - Utilize the Immediate window to make changes to the array while the program is running.
  - Monitor the change in the array using the Locals window. Note how all the “different” arrays behave when you change one of them.

## Submission Instructions

Submission should be made using a zip file that contains all of the Visual Studio C# project files. You will need to **zip the entire project folder** along with the .sln and .suo files. The folder will automatically contain the class source files as well as the executable file that is generated in \ProjectName\bin\Debug folder. Upload file to the Blackboard assignment drop box.

Zip the entire top-level folder by right-clicking the folder and selecting Send to | Compressed (zipped) folder.



Using built-in windows zip tools: <http://windows.microsoft.com/en-us/windows/compress-uncompress-files-zip-files>

Make sure you check the following. Your grade is dependent on all these criteria being met.

- You have included your name as a comment within your class.
  - e.g. “ // Inclass 4, Jane C. Smith, CIS 345, Tuesday 9:00 AM”
- Class file is called AverageCalculator.cs (rename from Program.cs).
- Your Visual Studio project is called AverageCalculatorApp.
- Zip filename is: **FirstNameLastName\_Inclass3.zip**

Verify your zip file before you submit

- Check for actual class files being present in the folder before you zip it.
- Check your zip file size after zipping – if it is 1K, it likely contains only a shortcut.
- Uncompress your zip file before submitting and verify that files are present.
- Download your zip file after submitting, uncompress, and again verify that your files are present.