

Class Averages

The goal of this lab is to give you practice programming with 2D arrays. You will write 2 functions for the ClassAverages.CPP program in order for it to work. Below is how the program will run once the functions have been written:

```
How many students? 3
How many assignments? 2
```

```
For student 1:
  Score 1: 95
  Score 2: 82
```

```
For student 2:
  Score 1: 79
  Score 2: 65
```

```
For student 3:
  Score 1: 89
  Score 2: 76
```

```
Student averages:
```

```
Student 1      88.5
Student 2      72
Student 3      82.5
```

```
Class averages:
```

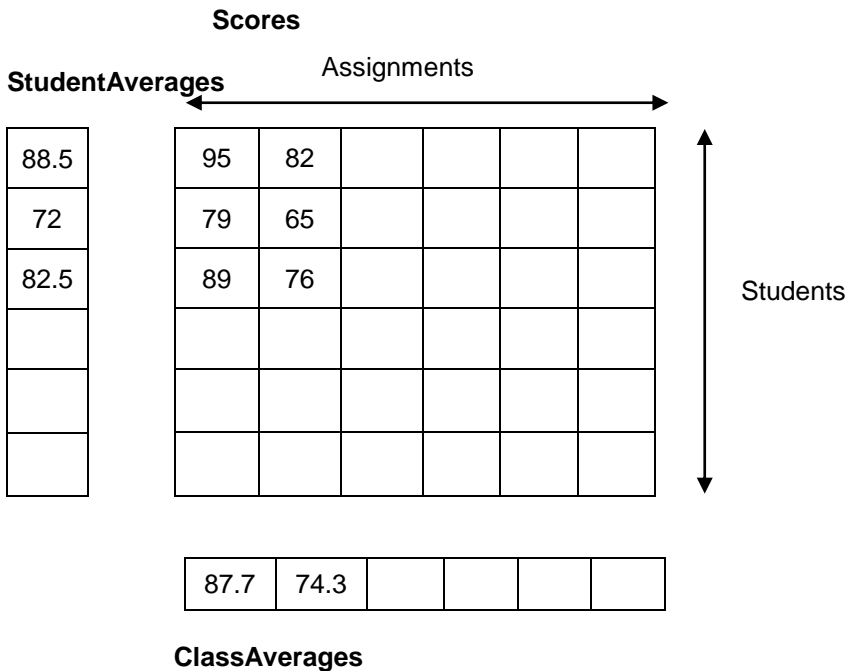
```
Assignment 1   87.6667
Assignment 2   74.3333
```

The two functions are:

FindStudentAverage - calculates the average score for each student

FindClassAveragesrage - calculates the average score for each of the assignments

This program will work with the 1D float arrays **StudentAverages** and **ClassAverages** and the 2D int array **Scores** all shown below. The **FindStudentsAverages** and **FindClassAveragesrages** functions will fill in the values in **StudentAverages** and **ClassAverages**.



The ClassAverages.CPP program is on the back of this sheet. The program can be obtained from http://www.harding.edu/DSTEIL/151/LABS/Class_Averages.cpp

Turn in a **printout** of the completed ClassAverages.CPP program. Make sure your name is at the top of the program.

```

/* ClassAverages.CPP
   This program accepts scores for a class and calculates student averages
   and assignment averages.
*/

#include <iostream.h>
#include <iomanip.h>

const int MAX_STUDENTS = 30;
const int MAX_ASSIGNMENTS= 5;

/* Function: FindStudentsAverages
   Receives all the scores and calculates the average for each student,
   placing the averages in the StudentAverages array.
*/

/* Function: FindClassAverages
   Receives all the scores and calculates the average for the entire class
   for each of the assignments, placing the averages in the ClassAverages array.
*/

void main()
{
    int Scores[MAX_STUDENTS][MAX_CURRENTASSIGNMENT];
    float StudentAverages[MAX_STUDENTS];
    float ClassAverages[MAX_CURRENTASSIGNMENT];

    int CurrentStudent;
    int TotalStudents;
    int CurrentAssignment;
    int TotalAssignments;

    cout << "How many students? ";
    cin >> TotalStudents;

    while (TotalStudents < 1 || TotalStudents > MAX_STUDENTS)
    {
        cout << "Please enter a number between 1 and " << MAX_STUDENTS << ": ";
        cin >> TotalStudents;
    }

    cout << "How many assignments? ";
    cin >> TotalAssignments;

    while (TotalAssignments < 1 || TotalAssignments > MAX_CURRENTASSIGNMENT)
    {
        cout << "Please enter a number between 1 and " << MAX_ASSIGNMENTS<< ": ";
        cin >> TotalAssignments;
    }

    // Read values

    for (CurrentStudent = 0; CurrentStudent < TotalStudents; CurrentStudent++)
    {
        cout << "\nFor student " << (CurrentStudent + 1) << ":\n";
        for (CurrentAssignment = 0; CurrentAssignment < TotalAssignments; CurrentAssignment++)
        {
            cout << "   Score " << (CurrentAssignment + 1) << ": ";
            cin >> scores[CurrentStudent][CurrentAssignment];
        }
    }

    FindStudentAverage(scores, StudentAverages, TotalStudents, TotalAssignments);    // Write this function

    cout << "\nStudent averages:\n";

    for (CurrentStudent = 0; CurrentStudent < TotalStudents; CurrentStudent++)
    {
        cout << "Student      " << (CurrentStudent + 1) << setw(10) << StudentAverages[CurrentStudent] <<
endl;
    }

    FindClassAverages(scores, ClassAverages, TotalStudents, TotalAssignments);    // Write this function

    cout << "\nClass averages:\n";

    for (CurrentAssignment = 0; CurrentAssignment < TotalAssignments; CurrentAssignment++)
    {
        cout << "Assignment " << (CurrentAssignment+1)<< setw(10) << ClassAverages[CurrentAssignment] <<
endl;
    }
}

```