\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**CSC253 C# ProGRAMMING**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

LAB 03 **METHODS: A DEEPER LOOK AND ARRAYS**

# Objectives

In this lab assignment, students will learn:

* How to generate random numbers
* How to create and use static methods
* How to use common Math class functions
* How to use optional parameters
* How to create and use recursive methods
* How to create and use one dimensional arrays
* How to create and use two dimensional arrays

# Goals

In this lab assignment, students will demonstrate the abilities to:

* Generate random numbers
* Create and use static methods
* Use common Math class functions
* Use optional parameters
* Create and use recursive methods
* Create and use one dimensional arrays
* Create and use two dimensional arrays

# Description

Create a C# console application for each question. When you create a new C# project, Visual Studio creates a folder to hold every file and sub-folder for your project. You need to zip this folder and submit the zip file to Blackboard.

1. Students in a course need to do 3 lab assignments and take 3 quizzes. Course grade is calculated from these scores. Write a program to calculate course grade for a student. You must write and use the following two functions.
2. Main function: Create two arrays to store lab scores and quiz scores. Use two loops to get these scores from the user. Tell the user that the default weights for labs and quizzes are 50 and 50, i.e., the average of labs and the average of quizzes each accounts for 50% of course grade. If the user wants to use the default weights, enter D. Otherwise, enter C. If the user chooses to use default weights, call the GradeCalculator function and pass the two score arrays as two arguments. Do not pass any other arguments. If the user chooses not to use default weights, ask the user to enter the weight for labs and the weight for quizzes, respectively. Then call the GradeCalculator function and pass the two score arrays, lab weight and test weight as four arguments.
3. GradeCalculator function: This function has four parameters: lab score array, test score array, lab weight and quiz weight. Lab weight and quiz weight are optional parameters. If no arguments are passed to them, their values are 50 and 50. First, calculate and display average lab score. Second, calculate and display average quiz score. Third, use average lab score with lab weight and average quiz score with quiz weight to calculate course grade. Display course grade.

The following is an example. Default weights are used.

Enter Lab 1 score: 85

Enter Lab 2 score: 77

Enter Lab 3 score: 90

Enter Quiz 1 score: 76

Enter Quiz 2 score: 84

Enter Quiz 3 score: 95

The default weights for course grade are 50% labs and 50% quizzes.

Enter C to change the weights or D to use default weights: d

Lab average: 84

Quiz average: 85

Course grade: 84.5

The following is another example. Default weights are not used.

Enter Lab 1 score: 85

Enter Lab 2 score: 77

Enter Lab 3 score: 92

Enter Quiz 1 score: 66

Enter Quiz 2 score: 98

Enter Quiz 3 score: 72

The default weights for course grade are 50% labs and 50% quizzes.

Enter C to change the weights or D to use default weights: c

Enter lab percentage (without the % sign): 60

Enter quiz percentage (without the % sign): 40

Lab average: 84.6666666666667

Quiz average: 78.6666666666667

Course grade: 82.2666666666667

1. This program uses recursive calls to find the largest value in a group of random numbers. You must write and use the following two functions.
2. Main function: Create an array to store 10 integers. Populate this array with random integers in the range of 1 to 99, inclusive. Call the Largest function and pass the array as an argument. Display the return value.
3. Largest function: This function receives an array as an argument. If the array has only one element, return that element. Otherwise, split the array into two halves. Call the Largest function twice with the two halves as the argument in the two calls, respectively. Use the Max function of the Math class to compare the two return values. Return the larger one.

The following is an example.

Array: 29 86 45 74 19 56 59 4 27 80

Largest number: 86

1. CSC888 has three students. They can take the final exam as many times as they want. Store their scores in a jagged two dimensional array. For each student, ask how many attempts that student has made and input the scores. After all scores of all students have been entered, display the scores and calculate averages in two ways.
2. Do the following for each student separately. Display the scores of all attempts. Calculate and display the average of these attempts. If an average cannot be calculated because a student has made no attempts, display a message about that.
3. Do the following for each attempt separately. Display the scores of all students who have made that attempt. Calculate and display the average of these scores. For example, suppose all 3 students have made attempt 1, display these three scores and their average. Suppose only student 1 and student 3 have made attempt 2, display these two scores and their average.

Example:

Student 1:

How many attempts has this student made? 2

Enter score: 97

Enter score: 88

Student 2:

How many attempts has this student made? 3

Enter score: 75

Enter score: 99

Enter score: 82

Student 3:

How many attempts has this student made? 0

Student 1 scores: 97 88

Average: 92.5

Student 2 scores: 75 99 82

Average: 85.3333333333333

Student 3 scores: This student has made no attempts.

Attempt 1 scores: 97 75

Average: 86

Attempt 2 scores: 88 99

Average: 93.5

Attempt 3 scores: 82

Average: 82

# Grading rubric

Program 1:

Main function [14 pts]

GradeCalculator function [14 pts]

Program 2:

Main function [14 pts]

Largest function [14 pts]

Program 3:

Jagged two dimensional array [14 pts]

Display scores and calculate average for each student [14 pts]

Display scores and calculate average for each attempt [16 pts]