

C Sharp Programming Exercise 02

Recursive Methods

C# Step by Step

This activity consists of four programming exercises. The following exercises are open book and open note. You are free to use any written documentation you wish. However, these are individual exercises, and you cannot consult with each other in writing your programs.

This programming exercise has four parts consisting of four requirements. The grade for each requirement is indicated, for a maximum of 100 points. At a minimum, your program must compile successfully and run.

A *recursive* function is a function that calls itself. Recursive activities are extremely common in every day life. When you mow your lawn, you cut one mower-width and see if you are finished. If so, you stop and drink a beer. If not, you cut one mower-width and see if you are finished, again and again. When you shop for groceries, you place an item in your cart and mark it off the list. If it's the last item, you check out. If not, you place an item in your cart and mark it off the list, again and again. Recursive functions work exactly the same way. You call the function, perform one task, and check to see if you are done. If so, you stop. If not, you call the function again. I have completed the first part of this exercise below, and you can see how simple this is.

Sum of numbers: 70 points Create a console application that will accept ten numbers between 0 and 100, and report their sum.

Average ten scores: 80 points Create a console application that will accept ten test scores between 0 and 100, average them, and report the numerical grade. For example, a teacher will input ten test scores and compute the average numerical grade.

Average a specific number of scores: 90 points Create a console application that will accept an arbitrary number test scores (as specified by the user) between 0 and 100, average them, and report a numerical grade for the average. For example, a teacher will input the total number of tests, then input the specified number of test scores and compute the average numerical grade.

Average a non-specific number of scores: 100 points Create a console application that will accept a number test scores (as calculated by the number of scores actually entered) between 0 and 100, average them, and report a letter grade for the average based on the usual scale. For example, a teacher will input any number test scores, and compute the average numerical grade and the letter grade. This part required you to program a *stop value*. You can choose any kind of stop value you want, typically stop values consist of "quit," "exit," Ctl-C, or Escape. I chose a negative one (-1) as a stop value, assuming that no student would ever score a negative grade on a test.

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace progex02
8 {
9     class Program
10    {
11        static void Main(string[] args)
12        {
```

```

13     Console.WriteLine("\nPart 1, sum 10 numbers.");
14     int sum = SumTenInts(0, 0);
15     Console.WriteLine($"The sum of ten integers is {sum}");
16
17     Console.WriteLine("\nPart 2, average 10 numbers.");
18     double avg = AvgTenInts(0, 0);
19     Console.WriteLine($"The average of ten integers is {avg}");
20
21     Console.WriteLine("\nPart 3, average user predetermined number of scores.");
22     Console.Write("How many scores do you wish to enter? ");
23     string noScores = Console.ReadLine();
24     int numScores = int.Parse(noScores);
25     double avg1 = AvgUnkInts(0, 0, numScores);
26     Console.WriteLine($"The average of {numScores} integers is {avg1}");
27
28     Console.WriteLine("\nPart 4, average non-predetermined number of scores.");
29     double avg2 = AvgAnyInts(0, 0);
30     Console.WriteLine($"The average of ten integers is {avg2}");
31 }
32
33 private static double AvgAnyInts(int sum, int count)
34 {
35     /*some code*/
36 }
37
38 private static double AvgUnkInts(int sum, int count, int numScores)
39 {
40     /*some code*/
41 }
42
43 private static double AvgTenInts(int sum, int count)
44 {
45     /*some code*/
46 }
47
48 private static int SumTenInts(int sum, int count)
49 {
50     Console.Write("Enter a score: ");
51     string input = Console.ReadLine();
52     sum += int.Parse(input);
53     count++;
54     if (count < 10)
55         return SumTenInts(sum, count);
56     else
57         return sum;
58 }
59 }
60 }

```