SAMPLE LAB REPORT

LAB 2
SECTION 2

SUBMITTED BY:

GREG RICE

SUBMISSION DATE:

8/26/2006

Lab Problem

The purpose of this lab is to create a simple program which contains input and output commands that will calculate the mean of six test scores. The program must allow the user to enter each test score and then display the average as the result. The objective of this lab is to learn how to use basic I/O and arithmetic operations.

Analysis

The problem states that we must compute the average of a series of six numeric test scores provided by the user. Therefore the problem input consists of six separate decimal numbers in the range of 0 to 100 and the problem output is the decimal average of those inputs. The only relevant formula otherwise required is $(s_1 + s_2 + \ldots + s_6)/6.0$ to compute the average.

Design

Our problem was to create a simple program designed to perform statistical operations. Subdividing the problem into a series of steps, our lab team decided on the following initial algorithm:

1.) Get the six test scores.
2.) Compute the average of the scores using the formula $(s_1 + s_2 + \ldots + s_6)/6.0$.
3.) Display the average.

In addition, we decided to extend our solution by asking the user to sign in with his or her initials. Once entered, the program would display his or her initials and ask the user to enter six separated test scores. Finally the program would calculate the mean of those scores and print the average to screen.

Using the basic outline above, we designed our program in steps. We first began be selecting the variable data types. All numeric data was assigned to `float` data types in order to accurately store decimal numbers in the range of 0 to 100, while initial would be stored to `char` types. We next modified our source code to utilize the `scanf()` command to process each input variable. To process and ensure proper input data, we utilized the hints given in the lab instructions.

Testing

In order to verify the results of the solution, we first entered a series of six identical numbers. Of course the average of any identical numbers is that same number. However our program failed to display the correct result. We found the problem to be with input of user initials. A `getchar()` call is needed after the user enters initials to remove the ‘enter’ key press from the computer’s input buffer. With the problem identified, we once again proceeded to test the program with six identical test scores. On the second try, the program correctly displayed the average. We then tested the program with a series of 5 other random cases, each time comparing the result to our hand calculators.

Comments

In doing this lab, I learned that one must always double check everything. For example, our program was not compiling. The problem was a simple as double checking to make sure there were semicolons and parenthesis in all the correct locations. Our final problem, however, was that we did not read through the entire lab, and we missed the hints on the back. Therefore, we needed to alter the `printf()` commands in order for our output to be consistent with our example in the hint. I learned a lot about `getchar()` in this lab. Previously, I did not understand it's importance in relation to character input and the I/O buffer. I still require more practice with `scanf()`, as I had a lot of difficulty in scanning numeric input.

```
// Greg Rice
// Lab #2
#include “stdafx.h”

int _tmain(int argc, _TCHAR* argv[])
{
	// User initials
	char first, middle, last;
	// Six Score Inputs, Mean Output
	float score1, score2, score3, score4, score5, score6, average;

	// Allow user to enter initials
	printf(“Enter your initials:”);
	scanf(“%c%c%c”, &first, &middle, &last);
	getchar();

	// Enter Score 1
	printf(“\nEnter Test Score (1):”);
	scanf(“%f”, &score1);
	while ((score1 < 0.0) || (score1 > 100.0))
	{
		printf(“\nError! Number must be between 0 and 100. Please re-enter
		number”);
		scanf(‘%f”, &score1);
	}

	// Enter Score 2
	printf(“\nEnter Test Score (2):”);
	scanf(“%f”, &score2);
	while ((score2 < 0.0) || (score2 > 100.0))
	{
		printf(“\nError! Number must be between 0 and 100. Please re-enter
		number”);
		scanf(‘%f”, &score2);
	}

	// Enter Score 3
	printf(“\nEnter Test Score (3):”);
	scanf(“%f”, &score3);
	while ((score3 < 0.0) || (score3 > 100.0))
	{
		printf(“\nError! Number must be between 0 and 100. Please re-enter
		number”);
		scanf(‘%f”, &score3);
	}

	// Enter Score 4
	printf(“\nEnter Test Score (4):”);
	scanf(“%f”, &score4);
	while ((score4 < 0.0) || (score4 > 100.0))
	{
		printf(“\nError! Number must be between 0 and 100. Please re-enter
		number”);
		scanf(‘%f”, &score4);
	}

	// Enter Score 5
	printf(“\nEnter Test Score (5):”);
	scanf(“%f”, &score5);
	while ((score5 < 0.0) || (score5 > 100.0))
	{
		printf(“\nError! Number must be between 0 and 100. Please re-enter
		number”);
		scanf(‘%f”, &score5);
	}
```

```
        // Enter Score 6
        printf("\nEnter Test Score (6):");
        scanf("%f", &score6);
        while ((score6 < 0.0) || (score6 > 100.0))
        {
               printf("\nError! Number must be between 0 and 100. Please re-enter
               number");
               scanf('%f", &score6);
        }
        getchar();

        // Compute Average and Display Output
        average = (score1 + score2 + score3 + score4 + score5 + score6) / (6.o);
        printf("\nTest Score Average for %c%c%c= %.2f \n", first, middle, last average);
        return 0;
}
```