Quest 12

300 EXP

CPSC121 SI

Erin: Congratulations on reaching level 5.

Erin: This time we will be going over functions. Functions are simply defined as a collection of statements to perform a specific task. Of course that's quite formal and doesn't really explain why we use them. Functions are used in modular programming, which is when you break up a program into smaller, manageable functions (also known as modules).

Erin: This helps cut down clutter, simplifies the process of writing programs, and improves maintainability of the programs.

Erin: With functions there are parts that you must be aware of. There are function calls, function definitions, and function prototypes.

Erin: The function call is the statement that causes the function to execute. The function definition is the code that makes up a function. The function prototype is used to tell the program which functions will be defined and used by the program.

Erin: The purpose of a function is whatever you want it to be. Functions are user-defined and will behave however you program to. So why don't we start with the different parts of a function definition. The function definition includes a function header and a function body. The function header is made up of a return type, name, and parameter list. The function body has… well the body. Here is an example:

```
int findSum(int x, int y, int z)
{
  int sum = x + y+ z;
  return sum;
}
```

Erin: This is quite simplistic, but it shows all the parts of the function definition.

Return type: int
Name: findSum
Parameters: int x, int y, int z
Body: The code inside block

Erin: Now that we know how to define functions, let's discuss calls and prototypes. Calls are when the function is actually executed. These are typically in the main, but user-define functions can also call other user defined functions… or itself even.

Erin: Function prototypes are made up of just the function header. These are put on the top of the program, before the main. As you know, computers execute programs top to bottom, so if the main or any function tries to call a function that is defined below it, the program will crash since it does not know what that function is yet. The work around is function prototypes. These will let the program know which functions are defined, so if the program ever tries to call these functions, it will then go look for the definition.

Erin: Alright that was an earful. Let's put this knowledge to use. This quest will be a joint effort between you and Rex. Rex has already started on the project and created a skeleton for you to work off of. He has put in all the prototypes and the main. It is up to you to define the functions to get this to work. I'll let Rex explain to you what you need to do.

Rex: Our program deals with a contest where contestants are judged by three judges. We must ask the user for how many contestants are participating. We then put ask for the three scores of each contestant X. X is the number assigned to the contestant. The number starts at 1 and goes up to the number of contestants. We need to find out which contestant has the maximum average score. Once we let the user know which contestant won and with what score, we need to ask the user if they want to run another contest.

Rex: Here is what the three functions need to do:

avg: calculates and returns the average of the 3 scores passed to it.

runContest: prompts the user for the scores of all the contestants. This function will call avg, and figure out which contestant has the highest average score. runContest will then call displayResults to tell the user who won and with what score.

displayResults: takes in the contestant number, and the average score of the contestant. Then it will display the results to the user.

Rex: The skeleton code can be found in the Quest Items/Quest_12 folder. It is called contestSkeleton.cpp