**Chapter#6: Setting-up and implementing void and value-returning functions –wk#10-12 for in class/lab work**

SAMPLE PROBLEM#1 – working with **void functions (keyboard input)**

Write a main function with three integer variables declared and appropriate input prompts to receive three input values from the keyboard. Write a **function prototype** (i.e. declaration) and a **function definition** for **FindMinimum** **(a void function),** that takes as input three integers and outputs the minimum value to the display screen. The main function should make a **function call** to **FindMinimum,** send three values to the function in order to display the minimum value.

SAMPLE PROBLEM#2 – working with **value-returning functions (keyboard input)**

Write a main function with three integer variables declared and appropriate input prompts to receive three input values from the keyboard. Write a **function prototype** (i.e. declaration) and **a function definition** for **FindMinimum (a value-returning function),** that takes as input three integers and returns the minimum value to the calling code. The main function should make a **function call** to **FindMinimum,** send three values to the function and get in return the minimum value to be later displayed in an output statement.

SAMPLE PROBLEM#3 – working with **void functions (file input)**

Write **a void function** **FindMinimum** that take as input a file of integers (of unknown length) and output to the screen the minimum value on the file. (Remember that all file parameters must be passed by reference.) In orderto process the file data and show output (the minimum value) …the main function should make a **function call** to **FindMinimum**. You must also create a data file Numbers.dat with the following numbers 67 87 23 103 ……..

SAMPLE PROBLEM#4 – working with **value-returning functions (file input)**

Write an **int** function **FindMinimum (a value-returning function),** that take as input a file of integers (of unknown length) and returns the minimum value on the file to the calling code. (Remember that all file parameters must be passed by reference.) In orderto process the file data and get the minimum value …the main function should make a **function call** to **FindMinimum .** and output to the display screen the minimum value returned from the function called. You must also create a data file Numbers.dat with the following numbers 67 87 23 103 ……..

SAMPLE PROBLEM#5 – more work with **void functions and File I/O operations**

Write a function **void OpenForInput (ifstream & someFile),** that is called from main with an incoming file stream parameter **OpenForInput(inFile); //function called from within main**

The function should prompt the user for the name of an input file and attempt to open the file.

Write another function **void GetData (ifstream& dataFile),** that may also be called from main directly after the prior call

**GetData(inFile); //function called from within main**

The function should read and echo print the contents of the file to the screen. Remember to close and clear the file stream once done with the reading.

Finally write another function **void CopyData (ifstream& dataFile),** that may also be called from main directly after the prior call

**CopyData(inFile); //function called from within main**The function should read and echo prints the contents of the file to the screen. The function should also open a new file for output and copy the contents of the original file onto this new file Remember to close and clear the file stream once done with the with the processing.

**Function specific Programming Challenges**

1. **Lowest Score Drop**

Write a program that calculates the average of a group of test scores, where the lowest score in the group is dropped. It should use the following functions:

* void getScore() should ask the user for a test score, store it in a reference parameter variable, and validate it. This function should be called by main once for each of the five scores to be entered.
* void calcAverage() should calculate and display the average of the four highest scores. This function should be called just once by main and should be passed the five scores.
* int findLowest() should find and return the lowest of the five scores passed to it. It should be called by calcAverage, which uses the function to determine which of the five scores to drop.

Input Validation: Do not accept test scores lower than 0 or higher than 100.

(Gaddis Chapter 6 Programming Challenge 11)

1. **High Scores**

Consider a text file named scores.txt that contains player scores for a game. A possible sample is shown below where Ronaldo's best score is 10400, Didier's best score is 9800, etc.

Ronaldo

10400

Didier

9800

Pele

12300

Kaka

8400

Cristiano

8000

Write a function named getHighScore that takes a string reference parameter and an integer reference parameter. The function should scan through the file and set the reference parameters to the name of the player with the highest score and the corresponding score.

1. **Reduce fractions**

Write a function named convertToLowestTerms that inputs two integer parameters by reference named numerator and denominator. The function should treat these variables as a fraction and reduce them to lowest terms. For example, if numerator is 20 and denominator is 60, then the function should change the variables to 1 and 3, respectively. This will require finding the greatest common divisor for the numerator and denominator then dividing both variables by that number. If the denominator is zero, the function should return false, otherwise the function should return true. Write a test program that uses convertToLowestTerms to reduce and output several fractions.

(Savitch Chapter 4 Programming Challenge 15)

1. **Sort numbers**

Write a function named sort that takes three integer parameters by reference. The function should rearrange the parameter values so that the first parameter gets set to the smallest value, the second parameter gets set to the second smallest value, and the third parameter gets set to the largest value. For example, given the variable assignments a = 30; b = 10; c = 20; then the function call sort(a,b,c) should result in a = 10, b = 20, and c = 30.

Note that the array construct will also give you a way to solve this problem for an arbitrary number of items instead of only for three items.

(Savitch Chapter 4 Programming Challenge 17)