**CS2010 Lab 9**

**Due: Friday, March 23, 2018**



Overview: This lab assignment is designed to give you practice working with functions that pass argument(s) by value and by reference. For this lab you are asked to work with a partner to create a Visual Studio C++ program to implement your solution to the problem.

Your task as organizers of the upcoming mile race is to determine whether runners qualify for entry given their age and best racing time from other mile races. The data file **lab9.txt** contains a name, age and best mile time (in seconds) for each runner. Here is sample data for the first two runners:

Mason

25

255

Layman

365

32

First create a new C++ project:

1. Use your last names and **lab9** for your project folder and .cpp file names (e.g., **JoynerJohnson\_lab9** and **JoynerJohnson\_lab9.cpp**). Include both names in a comment at the beginning of the .cpp file.

2. Copy the file **lab9.txt** from Canvas to your project folder.

Write a ***main*** function to do the following:

1. Declare an input file variable and variables to hold the racer's name, age and time.

2. Open the file **lab9.txt** and check the success of the file operation.

3. Use a while loop to read in the data from the file. Each time through the loop you should:

- Read in a name, age and time

- Display the racer's name, age and time all on one line on the screen

**Function 1: *checkAgeTime***

Once your program reads in and prints the data correctly, add the function ***checkAgeTime*,** described below, to your program. This function must use reference parameters to pass back its results (not a return statement). Remember, you will need a function prototype before the ***main*** function, a function call inside your ***main*** function (inside the while loop), and a function definition after your ***main*** function.

In the data file, it is assumed that the first number after a racer's name is the age and the second number is the time. Unfortunately, because of a communication error, some of the data was entered in the opposite order. The function ***checkAgeTime*** should figure out which data corresponds to each measure and switch the two numbers if necessary.

Call the ***checkAgeTime*** function in the loop after reading in a racer's data, passing the age and time as the arguments. If the age value is greater than the time value, assume they were entered incorrectly and switch them. For example, the age for Mason is 25 and the time is 255 seconds. Since the age is less than the time, assume the data is in order so no swap is needed. For Layman the "age" is 365 and the "time" is 32. Since the "age" is greater than the "time", assume the data is out of order and switch the two values.

*(How do you swap the values in two variables?)*

**Function 2: *convertTime***

Once the age and time are in the right order, add another function ***convertTime*** to convert the time from seconds to minutes and seconds. This function should have three parameters, the time read from the file and the minutes and seconds once the time is converted. For example, if the time passed to the function is 255 the function should set the minutes parameter to 4 and seconds to 15. For each parameter, decide what type of parameter passing should be used. Do not use any return statements in this function. The ***main*** function should print the values passed back. For example:

Name Age Time Time (min/sec)

--------------------------------------------------------------

Mason 25 255 4:15

Layman 32 365 6:05

**BONUS (1 pt) - Function 3: *qualify***

For a bonus point, add another function ***qualify*** to determine if the runner is qualified in his/her age category. This function should use a ***return*** statement to return the code value indicated below for the qualifying age/time ranges shown. If the runner's age/time are not in the ranges shown, return the value 0 to indicate that the runner did not qualify.

Age Time Qualify code

Under 30 300 seconds or less 1

30 – 50 360 seconds or less 2

Over 50 420 seconds or less 3

Any other age and time combination 0

Your ***main*** function should check the value returned by this function and print "Qualified" on the line with the runner's data if the code is 1, 2 or 3. For example, output for the first two runners should look like this:

Name Age Time Time (min/sec)

------------------------------------------------------------------------

Mason 25 255 4:15 Qualified

Layman 32 365 6:05

**BONUS (1 pt) - Function 4: *qualifiersByAge***

For another bonus point, add a fourth function called ***qualifiersByAge*** to count the number of runners who qualified in each age category as well as the total number of runners (all age categories) who did not qualify. Pass the Qualify code from the previous function and variables representing the count of the number of qualifiers in each age category to be updated by this function. For each argument decide whether it should be passed by value or passed by reference.

After all runner's data has been read in and processed, the ***main*** function should print the number of qualifying runners by age category, the total non-qualifying runners and total runners overall.

Age Category Number of Runners

--------------------------------------------------------------------

Qualifiers Under 30 xx

Qualifiers 30 to 50 xx

Qualifiers Over 50 xx

Non-qualifiers (all ages) xx

--------------------------------------------------------------------

Total runners xx

**CS2010 Lab 9: Grading Rubric**

Due: Friday, March 23, 2018

\_\_\_ .cpp file turned in on Canvas, named with your last names.

1 Main function correctly reads in runner data from file and displays one line on screen with each runner's information

2 ***checkAgeTime*** function – parameters passed by value/by reference as needed, correct age and time passed back

2 ***convertTime*** function – parameters passed by value/by reference as needed, correct minutes and seconds passed back

1 **Bonus! *qualify*** function – parameters passed by value/by reference as needed,

correct qualifier code passed back, message "Qualified" displayed for runners

who meet qualifying criteria

1 **Bonus! *qualifiersByAge*** function – parameters passed by value/by reference as

needed, correct counts passed back and table showing counts by age displayed

**5 Total Points** (**7** with bonus)