**CISP 1020**

**Lab 3, C++ STL**

**General Description**

This application is a system that tallies race data. //It is written in a CodeLite environment with c++11 enabled

**Detailed Description**

In many foot races, each competitor wears an RFID tag. When the racer crosses a sensor, data is retrieved from the tag such as racer number, the current time and the sensor number. This application initially reads racer data from a text file that contains logged data for a race and computes statistics for each racer. It also detects the possibility of cheating under the following circumstances:

* If the racer misses a sensor (did the racer take a short cut?)
* If the speed between two sensors is greater than 14 miles per hour (did the racer get in a vehicle at some point?).

**You must use an STL vector of Racer objects.**

Create **at least** the following files.

* racer.h, racer.cpp
* main.cpp

**HINT:** It will be easier in the long run to create a sensor class (with functions that calculate whether two sensors are contiguous and the speed between two sensors, for example) and a time stamp class (with functions that calculate the time between two time stamps). If you create a Sensor class, each racer can have multiple sensors (so, the Racer class has a vector of sensors). Each sensor has exactly one timestamp (so the Sensor class has one TimeStamp object).

The racer class must have at least:

1. Private data: racer number (an integer), an unknown number of sensor data. Sensor data includes sensor number (an integer that starts at 0 for start time; the last sensor is the finish line), mile marker for the sensor (a double) and time stamp for each sensor (when the racer passed the sensor) with hour, minute, second and millisecond. (You may also want instance variables that holds a racer’s total race time, total race distance, total race speed.)
2. Default constructor
3. Copy constructor
4. Set and get functions for private data
5. A “get” function that computes and returns the racer’s total time to complete the race
6. Overloaded operator< (a racer is < another if the racer’s total race time is less than the other’s)
7. Overloaded operator=

**Use the sort function in the STL to sort racers by total race time.**

Input comes from a text file containing the race log with the following format with fields separated by semicolons. Get the name of the text file from the user. If the file doesn’t exist or is empty, printout an appropriate error message. The first line contains the gun time, when the race officially started. Racer start times (the first sensor) may be different since not every racer will cross the starting line at the gun time. There will be no errors in the input file. Times are in military time and do not span more than a day.

hour:minute:second:millisecond;total number of sensors;total number of miles in race

racer name;racer number;sensor number;sensor mile marker;sensor time[:more sensor info]

An example input file for a race of 9.3 miles (~15km), 4 sensors and 7 racers.

8:0:0:0;4;9.3

Paula Radcliffe;123;0;0;8:1:30:0;1;3.1;8:16:15:45;2;6.2;8:32:45:59;3;9.3;8:48:15:12

Geoffrey Mutai;1;0;0;8:0:0:0;1;3.1;8:14:39:0;2;6.2;8:29:33:0;3;9.3;8:43:12:2

Jane Smith;111;0;0;8:12:1:0;1;3.1;8:40:30:0;2;6.2;9:7:45:9;3;9.3;9:31:3:25

John Doe;456;0;0;8:9:12:9;1;3.1;8:33:45:0;2;6.2;8:57:59:0;3;9.3;9:21:30:30

Slow Moe;432;0;0;8:15:0:0;1;3.1;9:00:0:0;2;6.2;9:45:0:0;3;9.3;10:30:0:0

Ms. Cheater;987;0;0;8:0:0:0;1;3.1;8:7:23:0;3;9.3;8:21:45:1

Mr. Cheater;6789;0;0;8:1:0:12;1;3.1;8:3:45:0;2;6.2;8:9:32:1;3;9.3;8:15:32:15

**HINT:** to parse C-strings, try using the *strtok* function to “tokenize” the C-string (<http://www.tutorialspoint.com/c_standard_library/c_function_strtok.htm>). For C++ string objects, you can use *find* and *substr* functions in the string class (<http://www.cplusplus.com/reference/string/string/?kw=string> ).

Output to the screen a table that is sorted from best time to last with an asterisk beside the names of racers whose data is suspect. Names will be no more than 30 characters. Racer numbers no more than 6 digits. Print the time in the format of HH:MM:SS:mmm

Name Number Time

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

\*Mr. Cheater 6789 00:14:32:003

\*Ms. Cheater 987 00:21:45:001

Geoffrey Mutai 1 00:43:12:002

Paula Radcliffe 123 00:46:45:012

John Doe 456 01:12:18:021

Jane Smith 111 01:19:02:025

Slow Moe 432 02:15:00:000

**Due Date (see D2L).** Rename your CodeLite folder to c1020axxLab6 (***substituting your username for c1020axx***) and copy it to directory N:\CSIT\1020P01\handin. This folder is NOT on PS11, but can be accessed from any PC in any lab on any campus. It can NOT be accessed from home, so make sure you manage your time such that you can turn in you lab in time from school.