CSE 437 REAL TIME SYSTEM ARCHITECTURES

HOMEWORK 1 – Thread Safe Set (Due 28th November 2022)

A. The requirements of the Set data structure are listed below:

1. The set shall accept a type template parameter that represents the type that will be stored in your set. That is, your set shall be instantiated with the following declaration:
   ThreadSafeSet<CustomType> myset;
   The set shall require the custom type to overload the following operators:
   bool CustomType::operator<(const CustomType &rhs) const;
   bool CustomType::operator==(const CustomType &rhs) const;
2. The set shall enable insertions:
   i. The set shall accept l-value references during insertions and make a copy of the reference to be inserted into the set using the custom class copy constructor/assignment operator
      CustomType mydata(<constructor params>);
      myset.insert(mydata);
   ii. The set shall accept r-value references during insertions and move the object into the set using the custom class move constructor/move assignment operator
      myset.insert(CustomType(…));
      or
      CustomType mydata(<constructor params>);
      myset.insert(std::move(mydata));
3. The set shall not permit insertion of an element that is equal to an existing element in the set. The insertion function shall return true for successful insertions, and false for elements that already exist in the set.
4. The set shall permit searching elements in the set. The search function shall return true for an input that equals to one of the existing elements in the set. Otherwise, the search function shall return false.
5. The set shall enable deletion of elements. The deletion function shall return true for successful deletions, and false for deletion attempts of non-existing elements.
6. The set shall provide a function to clear itself (that is, delete all its elements)
7. The set shall provide a function to report its size (that is, the number of elements stored in the set)
8. The set shall enable iteration of its element in ascending order using a lambda function as input. That is, upon the iterate call, the set shall call the “iteration code” for all its elements in ascending order.
   myset.iterate([](const CustomType &element) { // iteration code});
9. All the set functions shall be thread safe.
   i. Case 1: Your set will be accessible through one writer and oner reader thread concurrently.
   ii. Case 2: Your set will be accessible through multiple writer and multiple reader threads concurrently.

B. Your set class should be able to store custom structures that are constructed with or without multiple constructor parameters.

C. You are not allowed to use third party libraries (i.e., Boost).

D. Write two solutions to this homework considering requirement 9-Case 1 and Case 2.

E. Write two test programs that demonstrate the capability and performance of your two classes.

NOTES:

1. Your solutions will be evaluated according to their design, clarity, and performance (in terms of both space and time). A minimal solution using an STL set with a mutex will correspond to a minimal grade. Your classes should perform well during concurrent access with high contention.
2. Your test programs will also be graded; So, write your test programs with the same care as your set classes.
3. You will perform a full scale software engineering development cycle. We need all phases including problem definition, design, analysis, implementation, and testing documents. Provide all class diagrams.
   You will not get a good grade if you provide only implementation.
4. You must use Modern C++ language elements rather than OS specific functionality (i.e., Posix threads)
5. You will deliver your report (a pdf file), your source files together with a Makefile and nothing else. Deliver them in a single zip file, without object files, executables, debug files or any other binary file.