Midterm Makeup – Take Home Examination

To complete this makeup successfully you need to provide a Word document that contains two sets of things:

First, go over each of the questions on your examination and describe, in that document, what, if anything was wrong with your answer. If you received less than a score of 1 for a question, you should assume there is something wrong. This is intended to have you do a critical examination of each of your answers. You may refer to my solutions if you wish, but will get no credit for repeating any part of those answers.

The second thing you need to do is to answer, as completely as you can, all ten questions below. For questions like the first, after you finish implementing the code, please copy the source code into your word document.

I cannot accept your makeups after Monday, May 7, 2018. Please upload the document to the college server, CSE687 folder.

1. Implement a thread-safe vector that provides the same functionality as the std::vector, but supports safe read and write access by multiple concurrent threads.

2. Describe where you have used the four main principles discussed in class, LSP, OCP, DIP, and ISP, in your design of Project #4. You may wish to consult my solution of MT4Q2.

3. Please provide the definition of an asyncSelect method in the Query class that accepts a callable object to execute on each record of the DbCore<P> and a callback method that does something useful with the result. You may wish to look at my Query class which provides an asyncSelect method that accepts a single callable object and returns a std::future<Query>.

4. Explain what each of the initialization sequences is doing in the Compound Objects demo in Handouts/Repository/Cpp/CompoundObjects. Please do that for all the classes with initialization sequences in their code.

5. Why has the Cpp11-BlockingQueue made its copy operations =delete?. You can find that code in Handouts/Repository/Cpp/Cpp11-BlockingQueue.

6. Given the code fragments below, where the ellipsis (…) represents code that has not been disclosed to you:

   ```
   class X { … };  class Y { public: explicit Y(const X& x); … };
   ```

   what can you say about the compilation and execution of each of the following statements? Describe each of the operations that occur as this code executes.

   ```
   Y func(Y y) { … }
   X x;
   Y y = func(Y(x));
   ```

7. Suppose that objects of a class are required to correctly carry out copy and assignment operations. Support or refute the claim that the class must implement a copy constructor and assignment operator. You must consider all the cases (what are they?).

8. Use the std::list class to hold a collection of strings. Use algorithm(s) from the STL and a lambda to concatenate the strings.

9. What is the purpose of defining both an interface and an abstract class in an inheritance hierarcy?

10. Given the declaration:

    ```
      void X::mf(Y &y);
    ```

    How would you change the declaration to declare that the operation of this function will not change the state of the object invoking it? What can you do with the re-declared function that you could not do in its original form?