

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 8, 2017. Please upload the document to the college server, CSE687 folder.

1. Implement a thread-safe circular buffer, a data structure, to hold the last N instances of an unspecified type added to the buffer. For the first $N-1$ items the length of the buffer equals the number of added instances. After N insertions the buffer size is constant at N . Demonstrate that your design works.
2. What are the main design strategies used in Object Oriented Design?
3. Please provide the definition of a function that accepts an unspecified STL container and a callable function. When executed the function applies the callable object to each element of the container¹. Please do not use any of the existing STL algorithms for this question. Can you make this work for all of the STL containers or just some subset? Illustrate your answer with a lambda that displays information about each element of the container.
4. Explain what each of the initialization sequences is doing in the MTS17-MT1Q5 code demo. Please do that for all the classes with initialization sequences in their code.
5. For the Abstract Syntax Tree class from CodeAnalyzer, write a move constructor and a move assignment operator. Demonstrate that these operations work.

¹ This is essentially the behavior of the `std::for_each` algorithm.

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 hierarchy?
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?