Name: ______________________________________ ID: ____________

Here are some practice questions to help you study for the quiz.

1. What is the difference between a *has-a* relationship and an *is-a* relationship? What does that have to do with inheritance in C++?

2. What are some of the benefits of defining objects?

3. What are some benefits of inheritance?

4. Assume you have a `vector<float>`, write code using an iterator to calculate the number of values that are positive.

5. Define a new type `IntLinkedListIterator` which allows you to iterate over a linked list of int. Assume that the properties of each Node for the linked list are a `Node* next` and an `int value`

6. In addition to the operators we overload on assignment four for `ReviewCollectionIterator`, we also could overload the + operator. First, is this a good idea? That is, would it be misleading to imply *random access*? Second, in any case, provide an implementation of it so that + will update the position of the iterator from the assignment by *n* forward instead of just one. Do the same thing for your IntLinkedListIterator above. Is it a good idea with a linked list iterator?

7. Explain what the purpose of throwing an exception would be.

8. (Long, but useful practice) Define a templated type MyVector. Your templated type should maintain a collection of any amount. It should do this by maintaining a regular array with empty spots in it. Only certain spots will be considered part of the actual array. When the array is full, and you try to push_back an element, your class should double the size of the array and copy elements from the old array to the new one. If someone tries to get an element that is outside the range of the vector you are representing but inside the array you are storing, you should throw an exception.

9. (Long, but useful practice) Define an iterator that works on a MyVector. This will allow you to use the `begin()` and `end()` methods.