Some review questions to ponder. . .

- What's the difference between deep and shallow copy?
- In what order, when creative a derived class, are constructors called? Destructors?
- What does const do? What are some different ways to use it? What are the differences between these ways?
- What's the difference between static, dynamic, and automatic memory? Can you give an example of each?
- What is the 'this' keyword? How is it used? Why do we have it?
- What does they keyword 'virtual' do? When do we use it?
- What is the difference between an alias, or reference variable, and a pointer variable?
- How many difference constructors can we have in a class? How does the compiler know which one to call?
- What is a destructor? What does it do? Why is it needed?
- Why are destructors always virtual?
- What does 'operator overloading' mean? Why do we do it?
- What is a 'memory leak'? Give an example. What is a 'dangling pointer'? Give an example of that too.
- What happens when you don't specify a constructor and/or destructor in a class?
- What does O(N) mean? What is 'N'? If I had two algorithms, one of O(log N) and one of O(N), which would be 'faster'?
- What is dynamic dispatch? Give an example.

- What is the difference between a Tree, a Binary Tree, and a Binary Search Tree? Which one is more efficient to search? Why? Give your answers in big O notation.

- How does SelectionSort work? How does MergeSort work? QuickSort? What are the best and worse case running times of these algorithms?

- What are the operations for a Stack? What are the operations for a Queue?

- What does FIFO mean? FILO? Which of these applies to a Stack and which to a Queue?

- Compare an array and a linked list. Which is more efficient at adding an element? Removing an element? Express this in big O notation.

- If no default constructor is specified in a class, but some other constructor is, will the compiler write in a default constructor for you?

- What is the difference between the Is-A and Has-A relationships?

- What does the **: public <base class>** do on the end of a declaration of a derived class?

- When implementing a non-default constructor in a derived class, how do you specify which constructor you want called in the base class?

- What is static dispatch? How is it different from dynamic dispatch?

- What is an abstract class? Can you instantiate an abstract class? What are they used for?

- Give some examples of popular recursive algorithms. How would you use recursion to print out a linked list? How about printing the list backwards?

- Why don’t constants matter when talking about big O notation?

- Thinking in big O notation, what is the difference between a stand-alone variable notation and a loop that executes N times?

- Can a base class pointer point to an instance of a derived class? Why or why not? Can a derived class pointer point to an instance of a base class? Why or why not?