ECE 309 Exam 1 Review Guide

# Exam format

* 75 minute test
* 4 to 5 problems, similar to homework problems, possibly like these:
  + Implementing a linked list
  + Interpret the execution of C++ code or expressions using C++ language features we’ve studied so far
  + Describe/interpret the behavior of data structures, like lists, queue, stacks
  + Write a C++ program to solve a problem
  + Write example code to illustrate key topics, like constructors or public inheritance
* Bring something to write with. No notes, computers, or other materials will be allowed during the test.
* You will NOT be allowed to leave the classroom during the test; you must hand-in your exam if you want to go to the bathroom.
  + If you have a medical condition or disability, I will excuse you from this constraint if you provide me with appropriate documentation.
* I’ll provide a C++ reference sheet with all the C++ keywords and an ASCII table.

# Topics for review

1. C++ Language Features
   1. class, public, protected, private
   2. Using classes, calling methods, accessing members
   3. Default function parameters
   4. references
   5. public/private inheritance
   6. using
2. Class Constructors and Destructors
   1. Default Constructors
   2. Specialized Constructors
   3. Destructor
   4. When do constructors and destructors execute for local, global, and heap objects
3. Allocating and deallocating classes using new and delete
   1. Use new or new[] for heap allocation
   2. Use delete or delete[] for freeing heap memory
4. Runtime complexity
   1. Evaluate the big-O runtime complexity of operations on a linked list
5. Data structure
   1. Define data structure and identify an example of one
6. Linked Lists
   1. Describe a linked list or doubly linked list; describe common operations on linked lists.
   2. Implement basic operations on a singly- or doubly-linked list in C++:
      1. append to end
      2. append to head
      3. remove from head
      4. remove from tail
      5. remove from middle
      6. insert into middle
      7. empty
      8. traversal using an iterator
   3. Explain the big-O time complexity of operations on linked lists
   4. Explain the purpose of the iterator class; change how it works
   5. Draw a picture of a linked list as an operation is performed on it.
7. Queue
   1. Describe the operation of a Queue and common functions in its interface
   2. Interpret the execution of code manipulating a Queue
   3. Implement a queue in terms of a linked list
   4. Evaluate the big-O complexity of operations on a Queue
8. Stack
   1. Describe the operation of a Stack and common functions in its interface
   2. Interpret the execution of code manipulating a Stack
   3. Implement a stack in terms of a linked list
   4. Evaluate the big-O complexity of operations on a Stack
9. Separate a class into source file and header file.
10. Inheritance: Public Inheritance
    1. Explain the difference between a base class and a derived class
    2. Explain the protected keyword and what it’s used for.
    3. Describe the benefits of public class inheritance; is-a relationship
       1. Describe how public, protected, and private members of the base class treated in the derived class
    4. Define polymorphism and its properties
       1. Virtual Functions
       2. Derived Types are treated as base types
    5. Define Abstract Base Class
       1. What’s a pure virtual function?
       2. What is a feature of an abstract base class?
11. Private inheritance
    1. Describe the benefits of private inheritance; has-a relationship
       1. Describe how public, protected, and private members of the base class treated in the derived class
       2. Has-A relationship between base and derived classes
    2. Explain when private inheritance makes sense versus simple composition
    3. Explain the using keyword and its usefulness for re-declaring inherited class members.