C++ review material (my notes; also, see Sections 2.2-2.4, 3.3, 4.2)

- Very good knowledge of constructors, copy-constructors, and destructors).
- Very good knowledge of operator overloading (especially the assignment operator).
- Very good knowledge of pointers and dynamic array allocation.
- Differences between static and dynamic array allocation.

Section 2.6 (Comparison of Algorithms)

- Why do we need to compare algorithms?
- How do we compare algorithms?
- Asymptotic Analysis
- Big-O notation

Sections 5.1, 5.3, 6.1 (Array-based Stacks, Queues, and Templates)

- Very good knowledge of the Stack specification and implementation using arrays.
- Very good knowledge of the Queue specification and implementation using arrays.
- Implementation details (e.g., using "front" and "rear" to determine whether the queue is empty or full).
- Running times for all the functions of the Stack and Queue implementations.
- Good understanding of the "template" mechanism.

Sections 3.1, 3.2, 4.1, 4.2 (Unsorted Lists, Sorted Lists)

- Very good knowledge of the Unsorted List specification and implementation using arrays.
- Very good knowledge of the Sorted List specification and implementation using arrays.
- Implementation details (e.g., in the case of unsorted lists, it is more efficient to insert at the end of the list).
- Binary search algorithm (be careful when "item" is not in the list).
- Running times for all the functions of the Unsorted and Sorted List implementations.

Sections 5.2, 5.4 (Linked Stacks, Linked Queues)

- Very good knowledge of the Stack specification and implementation using linked-lists.
- Very good knowledge of the Queue specification and implementation using linked-lists.
- Running times for all the functions of the linked Stack and Queue implementations.
- Tradeoffs (e.g., memory or time) between array-based and linked-list-based implementations.

Sections 3.4, 3.5, 4.3 (Linked Unsorted List, Linked Sorted List)

- Very good knowledge of the Unsorted List specification and implementation using linked lists.
- Very good knowledge of the Sorted List specification and implementation using linked-lists.
- Running times for all the functions of the linked Unsorted List and Sorted List implementations.
- Tradeoffs (e.g., memory or time) between array-based and linked-list-based implementations.

Sections 3.4, 3.5, 4.3 (Variations of Linked Lists)

- Circular Lists - main idea
- Doubly linked-list - know how to insert/delete
- Singly vs doubly linked-lists - tradeoffs
- "Headers" and "Trailers" - main idea and benefits
- Implementing a linked-list using arrays - main idea and benefits

Programming Assignments

Them midterm might include 1-2 questions based on the material of the programming assignments (not coding). The purpose is to test your knowledge on the ideas and algorithms that we discussed in class for image manipulation.

General Comments

The midterm exam will be closed-books, closed-notes. The format will be similar to the sample midterm exams posted on the course's webpage (True/False, Short Answer Questions, Coding). Make sure you that you go over the examples we did in class. Also, do as many problems as you can from each chapter and review the questions in the quizzes and homeworks. Make sure that you understand the *trade-offs* between different implementations (e.f., array-based queues versus linked-list-based queues). Very important to know how to compute the running-time of an algorithm and how to express it in terms of big-O notation.