Exam 2 Review

The exam has 12 questions over the following content. It will be given in the testing center at JCCC over several days.

You can bring a calculator in order to do modulus operations - non-graphing only!

This exam does not have code-writing problems like the first exam.

Polymorphism

1. Use the lecture slides from http://edu.moosader.com/course/cs200/ for reference.
2. What is Polymorphism? What is its purpose?
3. How do you use Polymorphism in the code?
4. How are pointers used in Polymorphism?
5. How is Polymorphism related to a family of classes?
6. How much functions be declared in the family to ensure that the correct function is called when using Polymorphism?

Stacks and Queues Questions are similar to the Stack and Queue lab...

1. What are the common add, access, and remove functions for Stacks and for Queues?
2. What is "FIFO" and "LIFO", and which structures does each belong to?
3. Given a list of items, what is the result after Pop() for a Stack and for a Queue?

4. Draw a diagram of a Stack and a Queue after a series of Push/Pop calls.
5. Given a list of items, how many Pop() commands are needed to access an item, for both a Stack and a Queue.

Dictionaries

1. Be familiar with how to compute Modulus.
2. For some key and some TABLE_SIZE, be able to compute an index
3. Be able to identify when a generated index creates a collision
4. Be able to step through basic Linear Probing functionality (step forward by 1)
5. Be able to step through basic Quadratic Probing functionality (step forward by i^2)
6. Given a Hash and a Hash2 function, be able to calculate an index using Double Hashing.

Algorithm Efficiency

1. Use http://bigocheatsheet.com/ for studying from.
   (A Dictionary is also known as a Hash Table)
2. Know the average efficiency for...:
   - Adding an item to the end of an array
   - Adding an item to the end of a linked list
   - Adding an item to a dictionary
   - Accessing an item at some arbitrary position in an array
   - Accessing an item at some arbitrary position in a linked list
   - Searching through an unsorted array for a value
   - Searching through a linked list for a value