Test 1 will be Thursday, Oct. 3 in class. It will be closed-book and notes, except for one 8.5" x 11" sheet of paper (you can use front and back) containing any notes that you want AND the Python Summary handout. The test will cover the following topics (and maybe more).

Chapter 1. Introduction
Python control structures: if, while, for,
Python built-in data structures: list, dictionary, string
Preconditions, postconditions, and raising exceptions to enforce the precondition
Defining classes (e.g., Die) including inheritance, overriding methods

Chapter 2. Algorithm Analysis
Machine dependent measures of performance: program running time and instruction count
Machine independent measures of performance: big-oh, orders of complexity: constant O(1), logarithmic O(log n), linear O(n), "n log n"/log linear O(n log n), quadratic $O(n^2)$, cubic $O(n^3)$, exponential $O(2^n)$
Complexity analysis of an algorithm to determine its big-oh notation
Implementation of Python lists as an array of object references with implications on operation big-oh (e.g., pop() is O(1) while pop(0) is O(n), etc.)

Chapter 3. Basic Data Structures
General concept of a stack: LIFO, top and bottom
Stack Operations: pop, push, peek, size, isEmpty, __str__
Stack Implementations: Python list to store stack items and linked list of Nodes to store stack items including big-oh of operations
Stack Applications: general idea of using a stack to do parentheses matching and palindrome checking

General concept of a queue: FIFO, front and rear
Queue Operations: enqueue, dequeue, peek, size, isEmpty, __str__
Queue Implementations: Python list to store queue items and linked list of Nodes to store stack items including big-oh of operations

General concept of a deque: double ended queue, front and rear
Deque Operations: addFront, addRear, removeFront, removeRear, size, isEmpty, __str__
Deque Implementations: Python list to store deque items, singly-linked list of Nodes to store deque items, and doubly-linked list of Nodes (e.g, Node2Way) including big-oh of operations
Deque Applications: general idea of using deque to do palindrome checking

General concept of a list: head, tail, index
Categories of List operations: index-based, content-based, cursor-based
Unordered List operations and implementation with a singly-linked list of Nodes including big-oh of operations
Ordered List operations and implementation with a singly-linked list of Nodes including big-oh of operations

Section 6.6. Priority Queue with Binary Heaps
General concept of a priority queue: remove highest priority next
Priority Queue Operations: enqueue, dequeue, peek, size, isEmpty, __str__
Priority Queue Implementations: Python list unordered, Python list ordered by priority, Binary Heap including big-oh of operations
Binary Heap implementation: insert, findMin, delMin, isEmpty, size, buildHeap including big-oh of operations