Stacks and Queues

Stacks are a way of storing data. Visualize a stack of books. You can put one on top, take the top one off, and look at the bindings to find a book's location in the stack. To remove a book below the top, you would first have to remove the books on top of it. This is called "last in, first out" or LIFO. Stacks are commonly used in recursion, which will be covered later.

```
//creates a new stack
Stack<Double> stack1 = new Stack<Double>();
// uses the push method to put a new double on the top of the stack
stack1.push(1.0);   stack1.push(2.0);   stack1.push(3.0);
// uses the pop method to take off and return the top
double popped = stack1.pop();
//uses the peek method to return the top without removing it
double peeked = stack1.peek();
//returns the index of the first occurrence of the object if it is in
the stack
int searched = stack1.search(1.0);
```

In a queue you add items to one end and remove them from the other. Queue's store items in a First In, First out fashion, which also means items are removed in the same order they were added. Queues in Java can be compared to waiting in lines in real life because the first person in line is usually the first person to get what they came for. In Java the graphical user interface uses event queues to track all events, such as mouse and keyboard events.

```
//The LinkedList class implements the Queue interface
Queue<Integer> q = new LinkedList<Integer>();
//Adds to the tail of the queue
q.add(1); q.add(2); q.add(3);
// q is now [1,2,3]
//Removes the head of the queue
int head = q.remove();
//head is set to 1 and q is [2,3]
// gets the head of the queue without removing it
head=q.peek();
// head is set to 2
```