* Chapter 8
* Abstract data type
  + Data type whose properties (data and operations ) are specified independently of any particular implementation
* Views of data
  + Application (user) level
    - View of the data within a particular problem
    - View sees data objects in terms of properties (characteristics) and behaviors (actions)
  + Logical (abstract) level
    - Abstract view of the data and the set of operations to manipulate them
    - View sees data objects as groups of objects with similar properties and behaviors
  + Implementation level
    - Specific representation of the structure that hold the data items and the coding of the operations in a programming language
    - View sees the properties represented as specific data fields and behaviors represented as methods implemented in code
  + Describe word processor from three views
    - Font types
    - Point size
    - Paragraph properties
    - View
    - Review
    - Insert
    - References
    - Mailings
* Views of data
  + Composite data type
    - Data type given to a collection of values
  + Data structures
    - Implementation of a composite data field in a abstract data tye
  + Containers
    - Objects whose role is to hold and manipulate other objects
* Logical implementations
  + Array-based implementation
    - Objects in the container are kept in an array
  + Linked-based implementation
    - Objects in the container are not kept physically together, but each items tells you where to go to get the next one in the structure
* Stacks
  + An abstract data type in which accesses are made at only one end
    - LIFO, Last In First Out
    - The insert is called **Push** and the delete is called **Pop**
  + Are usually not recursive, unless they are doublely linked
* Queue
  + Abstract data type in which items are entered at one end and removed from the other
  + FIFO, first in first out
  + No standard queue terminology
* Stacks and queues are linked together by pointers
* List actions/operations that can be applied
  + Add item
  + Remove item
  + Get next item