1. **More Abstract**

Stacks, queues, and priority queues are more abstract entities than arrays and many other data storage structures. They're defined primarily by their interface: the permissible operations that can be carried out on them. The underlying mechanism used to implement them is typically not visible to their user. For example, the underlying mechanism for a stack can be an array, as shown in this

chapter, or it can be a linked list. The underlying mechanism for a priority queue can be an array or a special kind of tree called a *heap*.

1. **Stacks**A stack allows access to only one data item: the last item inserted. If you remove this item, then you can access the next-to-last item inserted, and so on.
2. **Efficiency of Stacks**

Items can be both pushed and popped from the stack implemented in the StackX class in

constant O(1) time. That is, the time is not dependent on how many items are in the stack, and is therefore very quick. No comparisons or moves are necessary.

1. **Queues**The word *queue* is British for *line* (the kind you wait in). In Britain, to "queue up" means to

get in line. In computer science a queue is a data structure that is similar to a stack,

except that in a queue the first item inserted is the first to be removed (FIFO), while in a stack, as we've seen, the last item inserted is the first to be removed (LIFO).