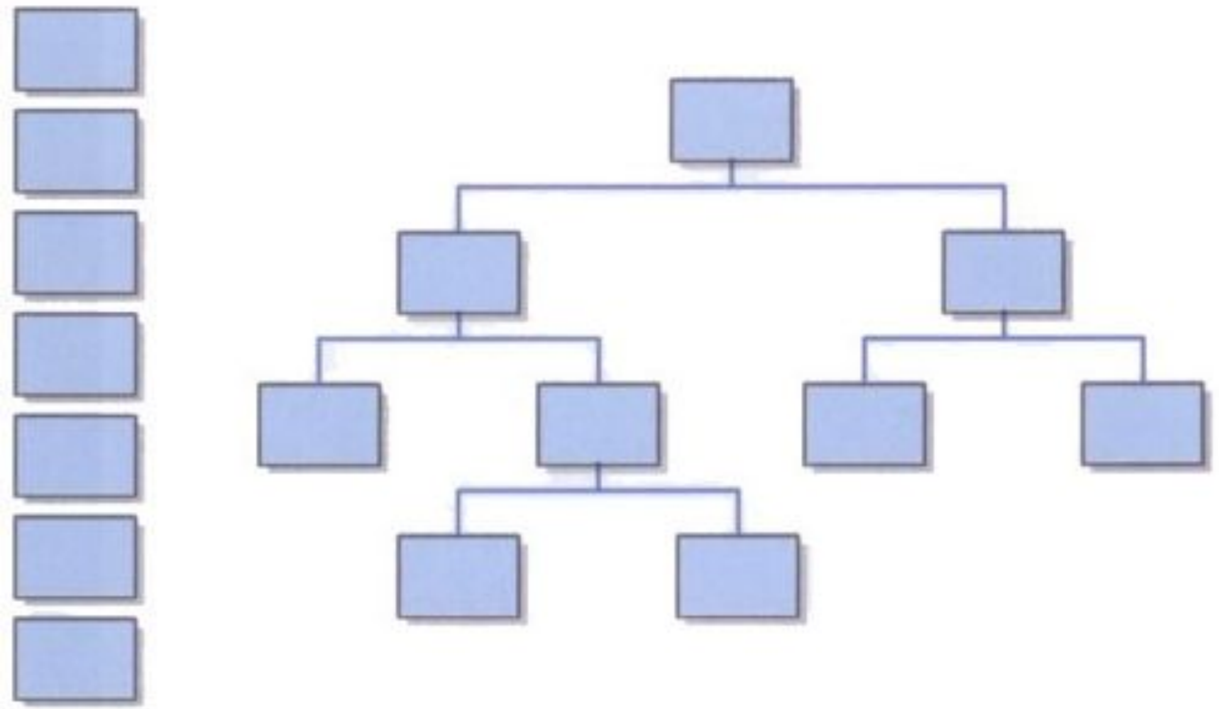
Collections

COMP128 Data Structures

Collections

- A collection is an object that holds and organizes other objects
- It provides operations for accessing and managing its elements
- Many standard collections have been defined over time
- Some collections are linear in nature, others are non-linear

Collections



Abstractions

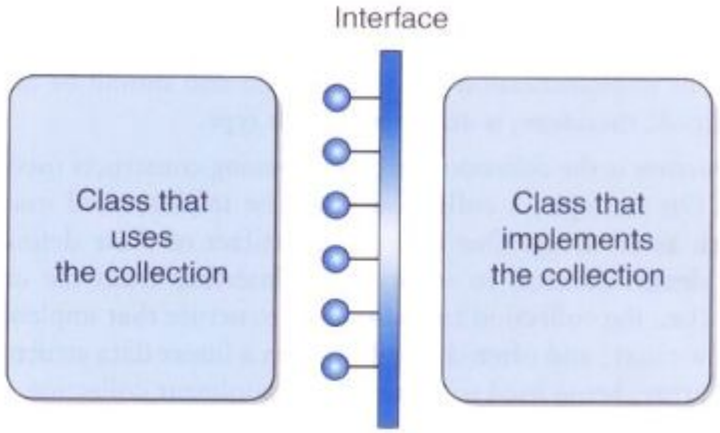
- An abstraction hides details to make a concept easier to manage
- All objects are abstractions in that they provide well-defined operations (the interface)
- They hide (encapsulate) the object's data and the implementation of the operations
- An object is a great mechanism for implementing a collection

Abstract Data Type

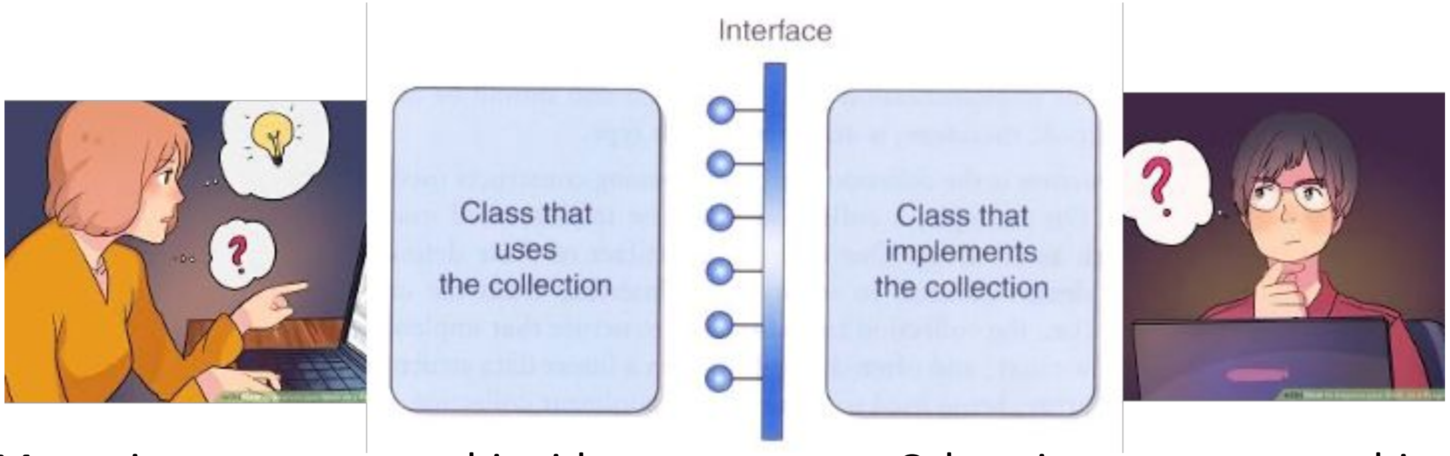
- A data type is a group of values and the operations defined on those values
- An abstract data type (ADT) is a data type that isn't pre-defined in the programming language
- A data structure is the set of programming constructs and techniques used to implement a collection
- The distinction between the terms ADT, data structure, and collection is sometimes blurred in casual use

Collection Abstraction

- A class that uses a collection interacts with it through a particular interface


Interface
Class that uses the collection
Class that implements the collection

Collection Abstraction


Interface
Class that uses the collection
Class that implements the collection

Many times you are on this side

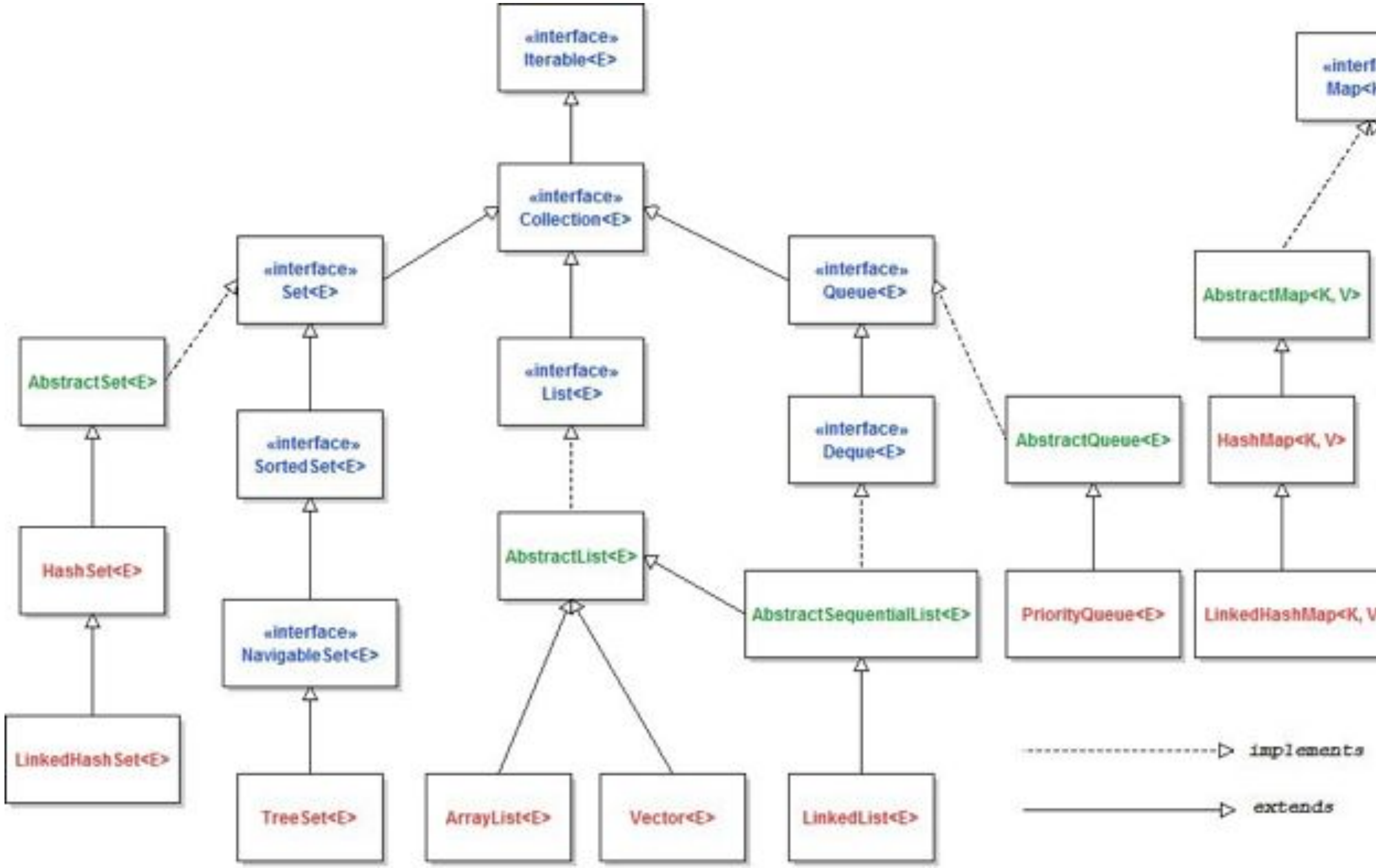
The programmer as **USER** of the collection

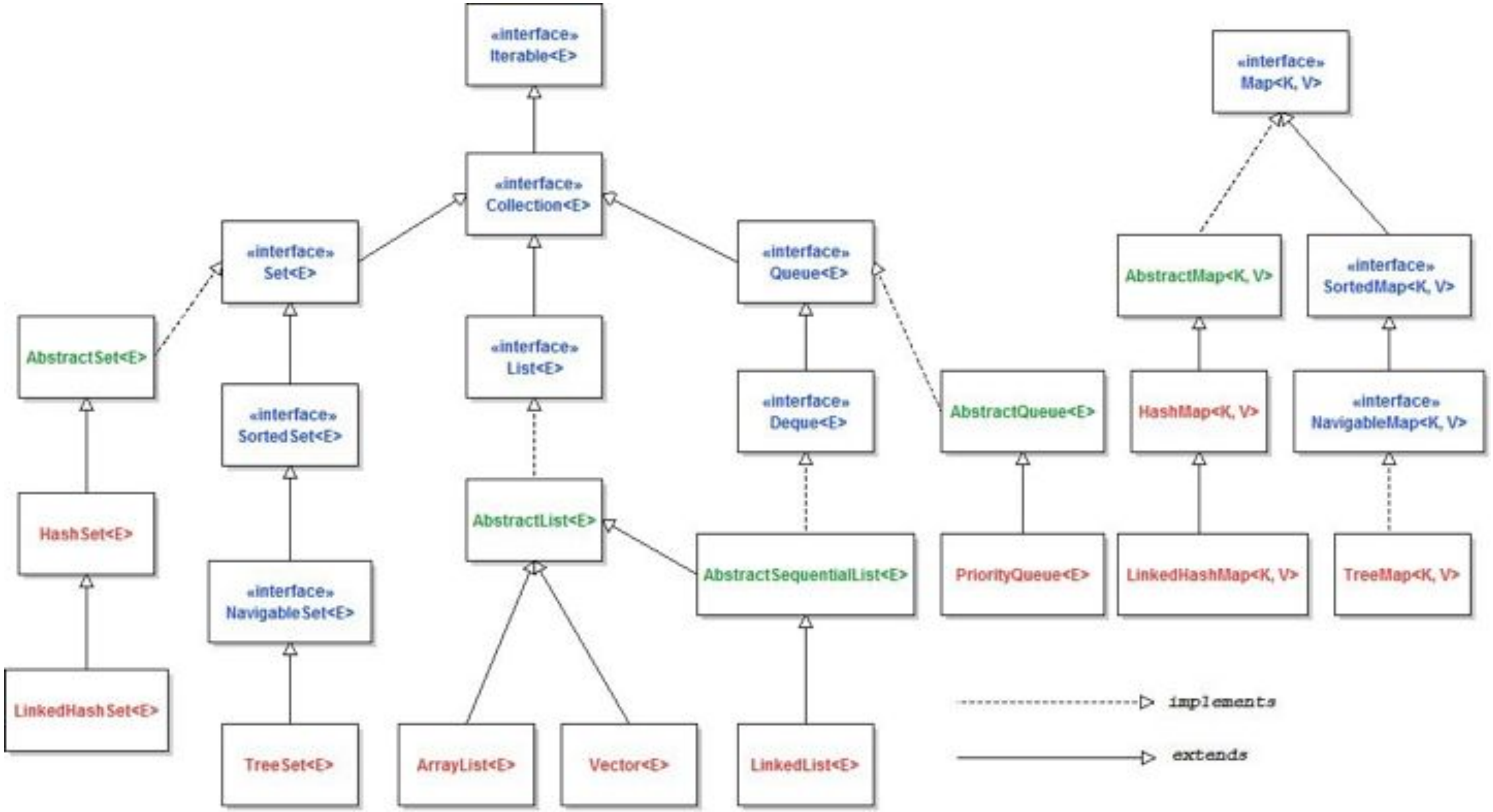
Other times you are on this side

The programmer as **IMPLEMENTER** of the collection

To use them well, you need some knowledge of how they are implemented. - That's us in this class!!

Collection Hierarchy





Collections Class & Arrays Class

Some operations on collections are pretty universal. To help simply things the Collections class and the Arrays class have static helper methods to do things like sort, fill, print, etc.

Today, we will be looking at the sort methods:

- Collections.sort(listVariable)
- Arrays.sort(arrayVariable)

Comparable Interface

Collections/Arrays.sort need to know how to order the items in the collection

The comparable interface is one way to define how objects are ordered.

Comparable Interface

To implement the interface, classes defining objects that will be stored in a collection implement the compareTo method:

int compareTo(T other)

// Returns a negative number (usually -1) if this should be ordered before other

// Returns 0 if this and other are equal

// Returns a positive number (usually 1) if this should be ordered after other

Reading Questions

(to be updated)

In-class Activity

Student Sorting Activity