# Holding Objects

## Prerequisite Topic: Generics

Generics allow type flexibility in code.
Generic type indicated with `<T>` or `<K, V>` usually
Compiler replaces generic types with given type (eg: `ArrayList<String>()`)

## Collections vs Maps

Collections hold a series of individual elements.
Maps store associated pairs of objects (called key-value pairs). The key object is used to look up the value object. Sometimes called dictionaries or associative arrays.
Both resize automatically (unlike Arrays).

## Collection Types

- List - Keeps a series of elements in the order they were added
- Set - Keeps a collection of unique elements
- Queue - Produces elements according to a "queueing discipline" (usually First-In-First-Out aka FIFO)

### List types

- ArrayList - Quicker with random access to elements
- LinkedList - Quicker with removal/insertion of elements in the middle of the list

### Set types

- HashSet - keeps elements in hash order (fast, not predictable)

- TreeSet - keeps elements in ascending sorted order
- LinkedHashSet - provides insertion-order access to elements

Sets are often used to test for membership using the `contains()` method

---

---

# Containers are Boxes

Whatever goes in eventually comes out according to certain rules. eg:

- Stacks: Last-in-first-out (LIFO)
- Queues: First-in-first-out (FIFO - usually)

---

# Iterators

Provides a way to iterate through a Collection.
Implicitly used in foreach loops.
Methods:

- `hasNext()` - returns true if there are more elements available
- `next()` - return the next element
- `remove()` - remove the last element returned from the underlying Collection

---

# ListIterators

Slightly more features than Iterators:

- Can iterate backward
- Provides index for previous and next elements

---

# Map types

- TreeMap - Keeps keys in insertion order
- HashMap - Hashes keys for quick access
- LinkedHashMap - Hashes keys, but preserves order with a LinkedList

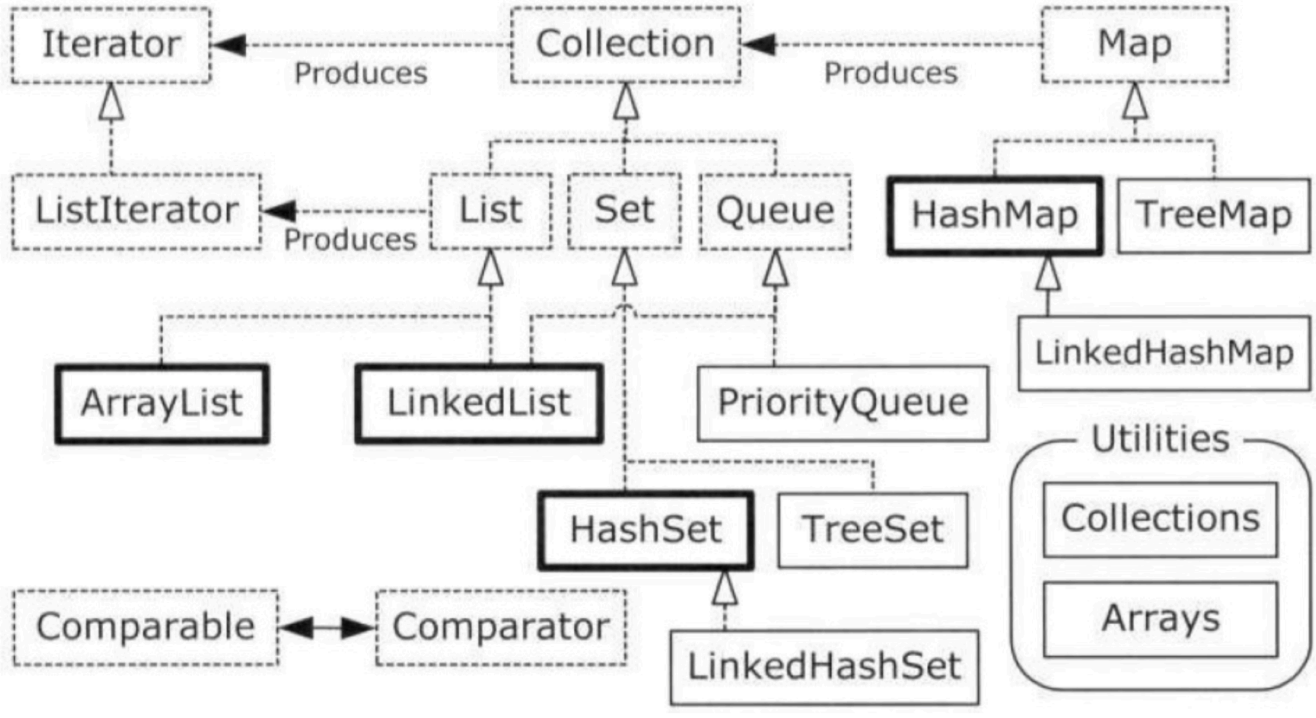
# Using maps

- Elements are pairs of objects, called keys and values
- keys are used to lookup values
- Check for a specific key with `conainsKey(key)`
- Add key-value pairs with `put(key, value)`
- Get values with `get(key)`
- Keys are stored in a Set, retrievable with `keySet()`

# Utility Classes

- Collections
- Arrays

### "Simple" Container diagram




**Simple Container Taxonomy**