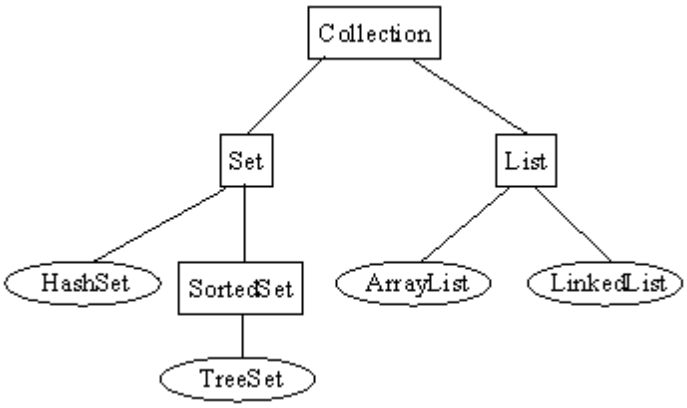
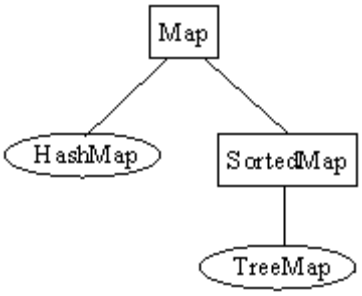
JAVA COLLECTIONS SUMMARY

CISC 124, fall 2001

(all classes & interfaces in `java.util`)





class ArrayList implements List

constructors:

```
ArrayList()
ArrayList(Collection c)
ArrayList(int initialCapacity)
```

interface Collection

```
// Basic Operations
boolean add(Object element);
boolean contains(Object element);
boolean isEmpty();
Iterator iterator();
boolean remove(Object element);
int size();

// Bulk Operations
boolean addAll(Collection c);
void clear();
boolean containsAll(Collection c);
boolean removeAll(Collection c);
boolean retainAll(Collection c);
```

class Collections

```
static Object max(Collection coll)
static Object max(Collection coll,
                  Comparator comp)
static Object min(Collection coll)
static Object min(Collection coll,
                  Comparator comp)
static void shuffle(List l)
static void sort(List l)
static void sort(List l, Comparator c)
```

interface Comparable

```
int compareTo(Object o);
```

interface Comparator

```
int compare(Object o1, Object o2)
```

class HashMap implements Map

constructors:

```
public HashMap()
public HashMap(Map t)
```

class HashSet implements Set

constructors:

```
HashSet()
HashSet(Collection c)
```

class LinkedList implements List

constructors:

```
LinkedList()
LinkedList(Collection c)
```

interface List extends Collection

```
// Positional Access
Object get(int index);
Object set(int index, Object element);
void add(int index, Object element);
Object remove(int index);
boolean addAll(int index,
               Collection c);
// Search
int indexOf(Object o);
int lastIndexOf(Object o);

// Iteration
ListIterator listIterator();
ListIterator listIterator(int index);

// Range-view
List subList(int from, int to);
```

interface ListIterator extends Iterator

```
void add(Object o);
boolean hasNext();
boolean hasPrevious();
Object next();
int nextIndex();
Object previous();
int previousIndex();
void remove();
void set(Object o);
```

interface Map

```
// Basic Operations
Object put(Object key, Object value);
Object get(Object key);
Object remove(Object key);
boolean containsKey(Object key);
boolean containsValue(Object value);
int size();
boolean isEmpty();

// Bulk Operations
void putAll(Map t);
void clear();

// Collection Views
public Set keySet();
public Collection values();
public Set entrySet();

// Interface for entrySet elements
public interface Entry {
    Object getKey();
    Object getValue();
    Object setValue(Object value);
}
```

interface Set extends Collection

no new methods

interface SortedMap extends Map

```
Comparator comparator();

SortedMap subMap(Object fromKey,
                 Object toKey);
SortedMap headMap(Object toKey);
SortedMap tailMap(Object fromKey);

Object first();
Object last();
```

interface SortedSet extends Set

```
// Range-view
SortedSet subSet(Object fromElement,
Object toElement);
SortedSet headSet(Object toElement);
SortedSet tailSet(Object fromElement);

// Endpoints
Object first();
Object last();

// Comparator access
Comparator comparator();
```

class TreeMap implements SortedMap

constructors:

```
TreeMap()
TreeMap(Comparator c)
TreeMap(Map m)
TreeMap(SortedMap m)
```

class TreeSet implements SortedSet

constructors:

```
TreeSet()
TreeSet(Comparator c)
TreeSet(Collection c)
```