

A+ Computer Science

SETS

Java Interfaces

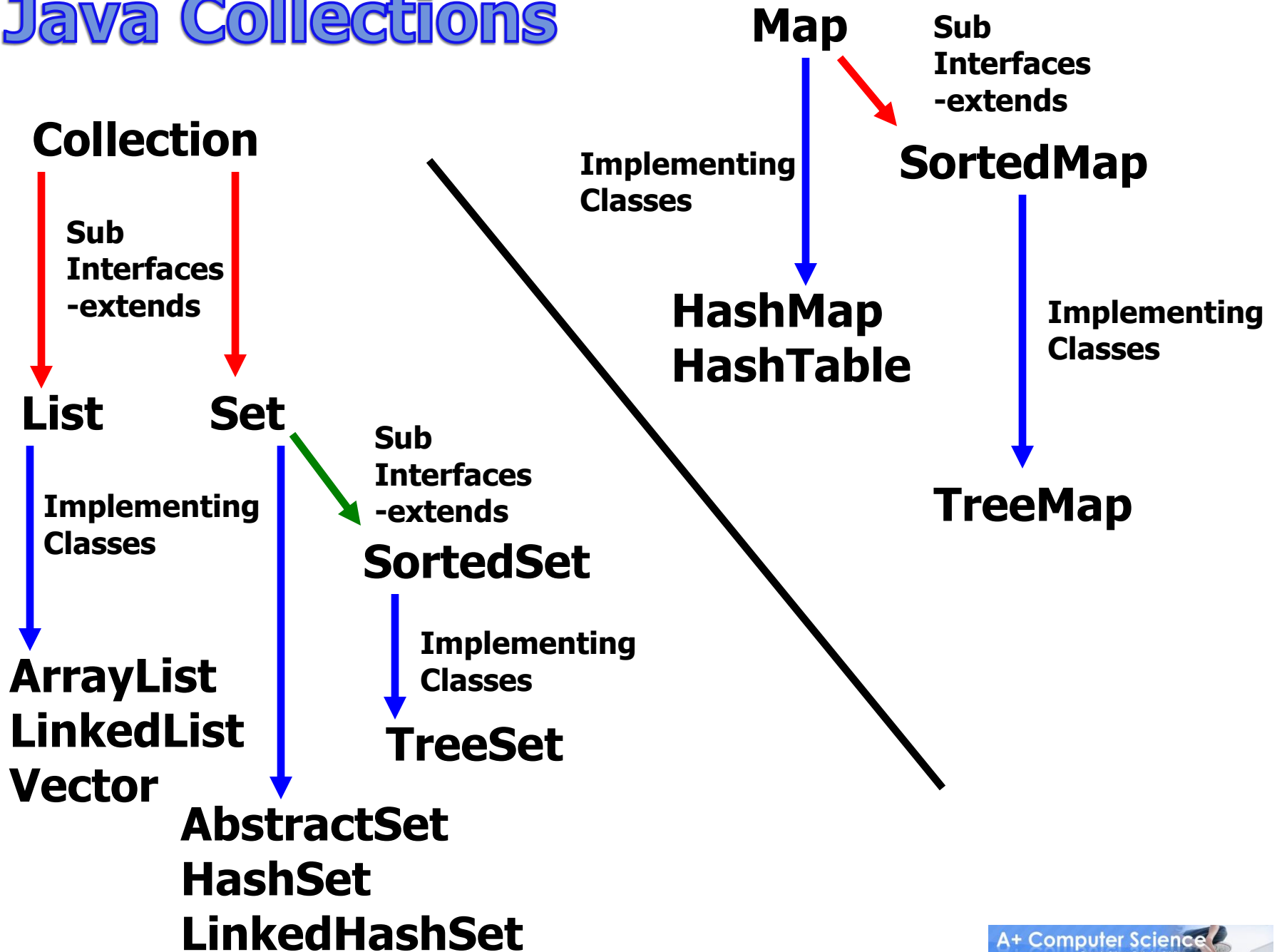
The following are important interfaces included in the Java language ::

Collection

Set

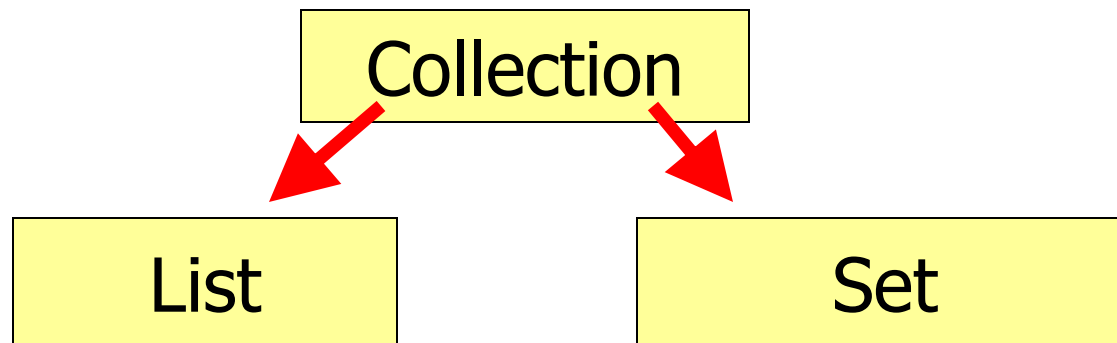
Map

Java Collections



Collection Interface

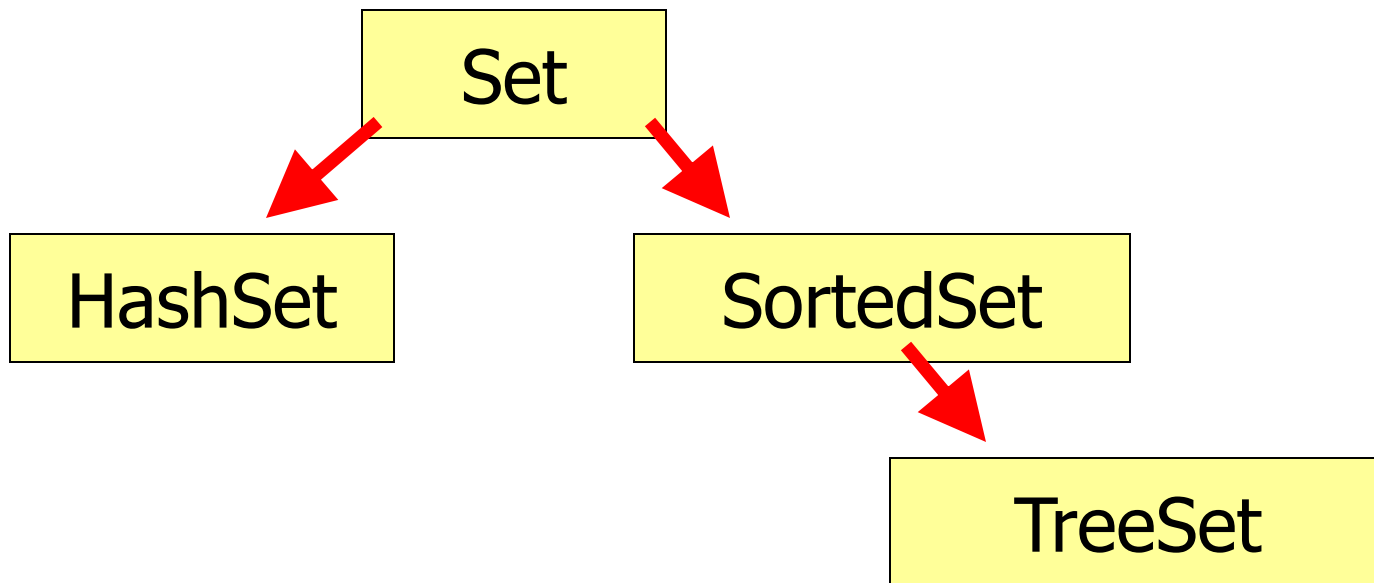
The Collection interface is the parent of List and Set. The Collection interface has many methods listed including add(), clear(), remove(), and size().



others not shown

Set Interface

The Set interface extends the Collection interface.



What is a set?

A set is a group of items all of the same type of which none are duplicates.

A set is very similar to an ArrayList.

Java Set

Because Set is an interface, you cannot instantiate it.

Set bad = new Set(); //illegal

Set hash = new HashSet(); //legal

Set tree = new TreeSet(); //legal

hash and tree store Object references.

Java Set

With Java 5, you can now specify which type of reference you want to store in the TreeSet or HashSet.

```
Set<Byte> bytes = new TreeSet<Byte>();  
Set<It> its = new HashSet<It>();
```


Java Set

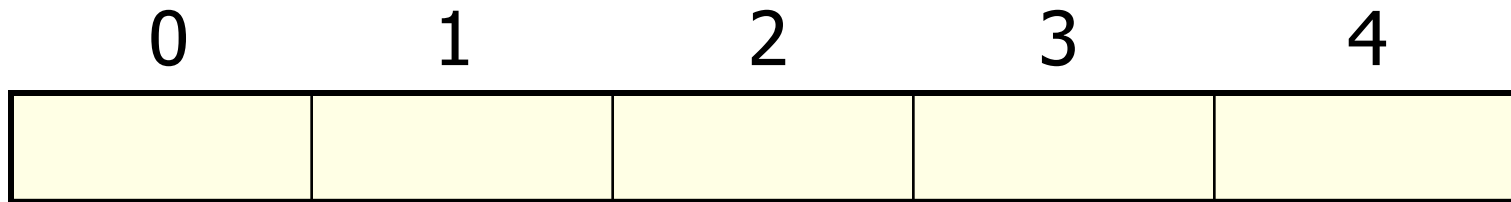
HashSet – a set ordered by each item's hashCode that is extremely time efficient.

TreeSet – a naturally ordered set that is very efficient, but not as efficient as HashSet.

What is a hash table?

HashSet and HashMap were both created around hash tables.

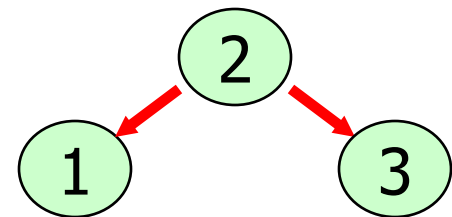
A hash table essentially a giant array. Each item is inserted into the array according to a hash formula.



What is a binary tree?

TreeSet and TreeMap were built around binary trees.

A Binary Tree is a group of nodes that contain left and right references. Each item is inserted into the tree according to its relationship to the other nodes.



Java Set methods

Set

frequently used methods

Name	Use
add(x)	adds item x to the set
remove(x)	removes an item from the set
clear()	removes all items from the set
size()	returns the # of items in the set

HashSet add()

```
Set<Integer> vals;  
vals = new HashSet<Integer>();  
vals.add(3);  
vals.add(6);  
System.out.println(vals.add(3));  
vals.add(-5);  
System.out.println(vals);
```

OUTPUT

false

[3, 6, -5]

HashSet add()

```
Set<String> vals;  
vals = new HashSet<String>();  
vals.add("applus");  
vals.add("comp");  
vals.add("sci");  
vals.add("sci");  
System.out.println(vals);
```

OUTPUT

[comp, applus, sci]

hashsetint.java
hashsetstring.java

HashSet remove()

```
Set<Double> vals;  
vals = new HashSet<Double>();  
vals.add( .3 );  
vals.add( 1.2 );  
vals.add( 2.6 );  
System.out.println(vals);  
vals.remove( 9.1 );  
vals.remove( 0.3 );  
System.out.println(vals);
```

OUTPUT

```
[2.6, 1.2, 0.3]  
[2.6, 1.2]
```

hashsetremove.java

TreeSet add()

```
Set<Integer> vals;  
vals = new TreeSet<Integer>();  
vals.add( 11 );  
vals.add( 3 );  
System.out.println(vals.add( 3 ));  
vals.add( 99 );  
System.out.println(vals);
```

OUTPUT

false

[3, 11, 99]

TreeSet add()

```
Set<String> vals;  
vals = new TreeSet<String>();  
vals.add("applus");  
vals.add("comp");  
vals.add("comp");  
vals.add("sci");  
System.out.println(vals);
```

OUTPUT

[applus, comp, sci]

treesetint.java
treesetstring.java

TreeSet remove()

```
Set<Double> vals;  
vals = new TreeSet<Double>();  
vals.add( .7 );  
vals.add( 1.1 );  
vals.add( 3.5 );  
System.out.println(vals);  
vals.remove( 2.2 );  
vals.remove( 0.7 );  
System.out.println(vals);
```

OUTPUT

```
[0.7, 1.1, 3.5]  
[1.1, 3.5]
```

treesetremove.java

TreeSet Output

```
Set<Double> vals;  
vals = new TreeSet<Double>();  
vals.add( 2.9 );  
vals.add( 3.5 );  
vals.add( 6.6 );  
  
Iterator<Double> it;  
it = vals.iterator();  
while(it.hasNext()){  
    System.out.println(it.next());  
}
```

OUTPUT

2.9

3.5

6.6

setoutput.java

TreeSet Output

```
Set<Double> vals;  
vals = new TreeSet<Double>();  
vals.add( 1.1 );  
vals.add( 0.4 );  
vals.add( 12.6 );  
  
for(double dec : vals)  
{  
    System.out.println(dec);  
}
```

OUTPUT

0.4

1.1

12.6

setoutputnew.java
setsplit.java

Work on Programs!

Crank

Some Code!

A+ Computer Science

SETS