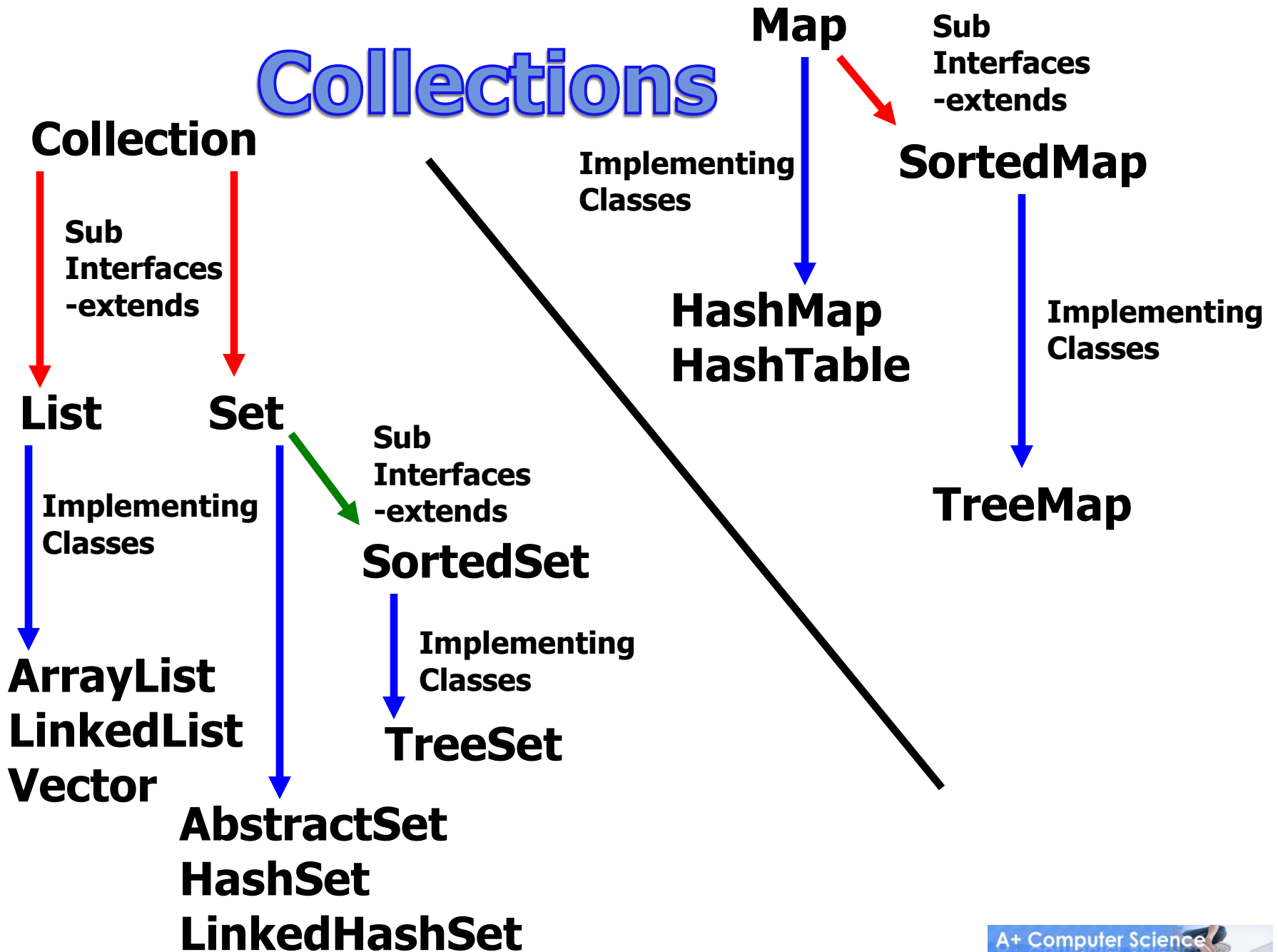


A+ Computer Science

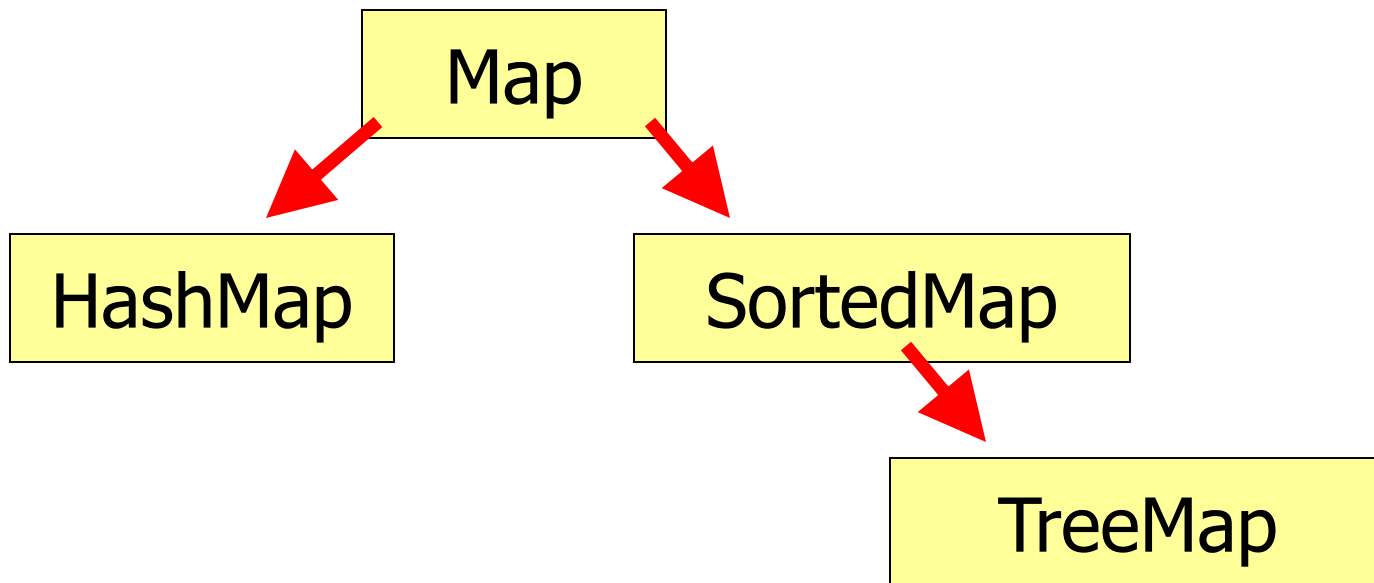
Java maps

Collections



Java Map Class

The Map interface does not extend any other interface.



Java Map Class

**A Map stores pairs of keys and values.
Each key – value pair is unique.**

**A translation program could be
written using a map.**

Maps cannot store duplicates.

Java Map Class

Key	Value
restroom	bano
cat	gato
boy	muchacho
house	casa
toad	sapo
water	agua

Java Map Class

Because Map is an interface, you cannot instantiate it.

Map bad = new Map(); //illegal

Map hash = new HashMap(); //legal

Map tree = new TreeMap(); //legal

hash and tree store Object references.

Java Map Class

With Java 5, you can now specify which type of references you want to store in the TreeMap or HashMap.

```
Map<String, Integer> hash;  
hash = new HashMap<String, Integer>();
```

```
Map<String, Set> tree;  
tree =  
    new TreeMap<String, TreeSet<String>>();
```

Java Map Class

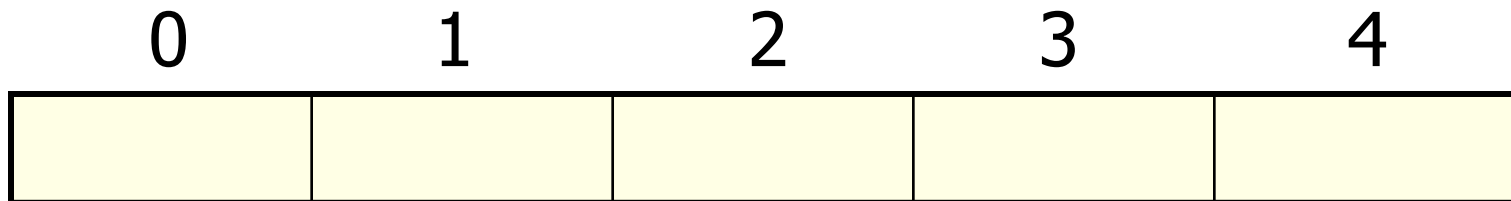
HashMap – a map ordered by each item's hashCode that is extremely time efficient.

TreeMap – a naturally ordered map that is very efficient, but not as efficient as HashMap.

Java Map Class

HashSet and HashMap were both created around hash tables.

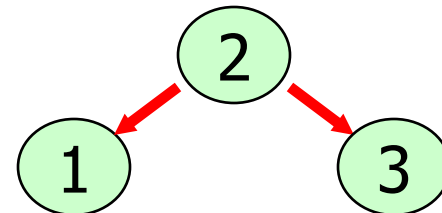
A hash table is a giant array. Each item is inserted into the array according to a hash formula.



Java Map Class

TreeSet and TreeMap were built around balanced 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.



Map Methods

Map

frequently used methods

Name	Use
<code>put(x,y)</code>	adds the <code><x,y></code> pair to the map
<code>get(x)</code>	gets the value for key <code>x</code>
<code>clear()</code>	removes all items from the set
<code>size()</code>	returns the # of items in the set
<code>keySet()</code>	returns a set of all keys in the map
<code>containsKey(x)</code>	checks if key <code>x</code> is in the map

Java TreeMap

```
Map<Integer,String> map;  
map = new TreeMap<Integer,String>();  
map.put(1,"aplus");  
map.put(2,"comp");  
map.put(3,"sci");  
map.put(4,"is");  
map.put(5,"the");  
map.put(6,"best");
```

```
System.out.println(map.get(1));  
System.out.println(map.get(5));  
System.out.println(map.get(7));
```

OUTPUT

**aplus
the
null**

Java TreeMap

```
Map<Integer,Double> map;  
map = new TreeMap<Integer,Double>();  
map.put(5,2.5);  
map.put(8,6.7);  
map.put(11,5.9);  
map.put(6,4.2);  
map.put(17,1.5);  
System.out.println(map.put(8,9.5));  
System.out.println(map.put(6,6.6));  
  
System.out.println(map.get( 6 ));  
System.out.println(map.get( 11));  
System.out.println(map.get( 7 ));
```

OUTPUT

6.7

4.2

6.6

5.9

null

Java HashMap

```
Map<Integer,Double> map;  
map = new HashMap<Integer,Double>();  
map.put(5,2.5);  
map.put(8,6.7);  
map.put(11,5.9);  
map.put(6,4.2);  
map.put(17,1.5);  
System.out.println(map.put(8,9.5));  
System.out.println(map.put(6,6.6));  
  
System.out.println(map.get( 6 ));  
System.out.println(map.get( 11));  
System.out.println(map.get( 7 ));
```

OUTPUT

6.7

4.2

6.6

5.9

null

basicmapone.java
basicmaptwo.java
basicmapthree.java

Map put() method

```
Map<Character,Integer> map;  
map = new TreeMap<Character,Integer>();  
  
String s = "apluscompscirockscoputerscience";  
for(char c : s.toCharArray())  
{  
    if(map.get(c)==null) c is not in the map.  
        map.put(c,0);  
    map.put(c,map.get(c)+1); bump up the count  
}  
System.out.println(map.get('a'));  
System.out.println(map.get('x'));  
System.out.println(map.get('c'));  
System.out.println(map.get('t'));
```

OUTPUT

1

null

6

treemapputone.java

Map put() method

```
Map<Character,Integer> map;  
map = new TreeMap<Character,Integer>();
```

```
String s = "cabdefghihabcdc";  
for(char c : s.toCharArray())
```

```
{  
    if(map.containsKey(c)) c is in the map.  
    {  
        map.put(c,map.get(c)+1);  
    }  
    else c is not in the map.  
    {  
        map.put(c,1);  
    }  
}
```

```
System.out.println(map.get('a'));  
System.out.println(map.get('x'));  
System.out.println(map.get('c'));
```

OUTPUT
2
null
4

treemapputtwo.java

Map Output

```
Iterator<Character> it;  
it = map.keySet().iterator();  
while(it.hasNext())  
{  
    char c = it.next();  
    System.out.println(c+" - "+map.get(c));  
}
```

Map Output

```
for(char c : map.keySet())  
{  
    System.out.println(c+" - "+map.get(c));  
}
```

treemapoutput.java
treemapoutputforeach.java

Output Map Values

```
for(double d : map.values())  
{  
    System.out.println(c);  
}
```

Key	Value
a	7.0
b	2.0
c	6.0

OUTPUT

7.0 2.0 6.0

treemapoutputvalues.java
hashmapoutput.java

Work on Programs!

Crank

Some Code!

A+ Computer Science

Java maps