**[Generics](https://docs.oracle.com/javase/tutorial/extra/generics/index.html):**

* Generics allow you to abstract over *types*
* <https://docs.oracle.com/javase/tutorial/extra/generics/intro.html>
* <https://docs.oracle.com/javase/tutorial/java/generics/index.html>
* Function overload with generic types (same number of params) is not allowed
  + public void foo(List<String> a);
  + public void foo(List<Integer> b);
    - Above both are not allowed
  + public void foo2(T a);
  + public void foo2(K b);
    - Above both are not allowed

[**Comparators**](https://docs.oracle.com/javase/tutorial/extra/generics/index.html)**:**

[**Arrays**](https://docs.oracle.com/javase/8/docs/api/java/util/Arrays.html)**:**

[**Java Modifiers**](https://www.tutorialspoint.com/java/java_modifier_types.htm)**:**

* **Java Access Modifiers**
* **Non Access Modifiers**
  + Transient
  + Volatile

[**Google Core Libraries for Java:**](https://github.com/google/guava)

**Cloneable, Serializable Interfaces**

<https://stackoverflow.com/questions/2889777/difference-between-hashmap-linkedhashmap-and-treemap>

Hash Function

* Good hash function disperses the elements properly among the buckets

[References in Java](http://www.baeldung.com/java-weakhashmap)

* Weak
* Strong
* Phantom
* Soft

How To Synchronize collection objects

* Set - Collections.SynchronizedSet(new HashSet())
* Map - Collections.SynchronizedMap(new HashMap())

The worst case search time for a *sorted* linked list is O(n)

More list/algorithms - <http://www.geeksforgeeks.org/skip-list/>

Problems & Algorithms - http://algorithms.tutorialhorizon.com/