**Java Collections**

**A Collection:**

A ***collection*** is a collection of objects that share some common relationship or purpose. Different types of collections have specific names and expected behaviors. These include arrays, lists, vectors, sets, queues, tables, dictionaries, and maps (Data structure that maps keys to values).

These collections are differentiated by how they store objects in memory, how objects are retrieved, and ordered, and whether they allow null values or duplicate entries.

**A collection Framework:**

Oracle’s Java documentation describes its collection framework as:

“A unified architecture for representing and manipulating collections, enabling collections to be manipulated independently of implementation details ”

***Its based on Interfaces***

**What’s in the framework, what’s not?**

Arrays and the Array utilities in the java.util.Arrays class are not considered part of this framework. A collection objects implements the Collection interface, with the exception of maps. The Collection interface is the foundation of the collection hierarchy in Java.

**The Collection Interface**

This interface is often used when you want to **pass collections** around and manipulate then with **maximum flexibility and generality**.

***collection interface doesn’t have sort()***

***List interface extends collection. Collection is a base interface***

**What’s a polymorphic algorithm**

Oracles’s documentation describes a polymorphic algorithm as a piece of reusable functionality.