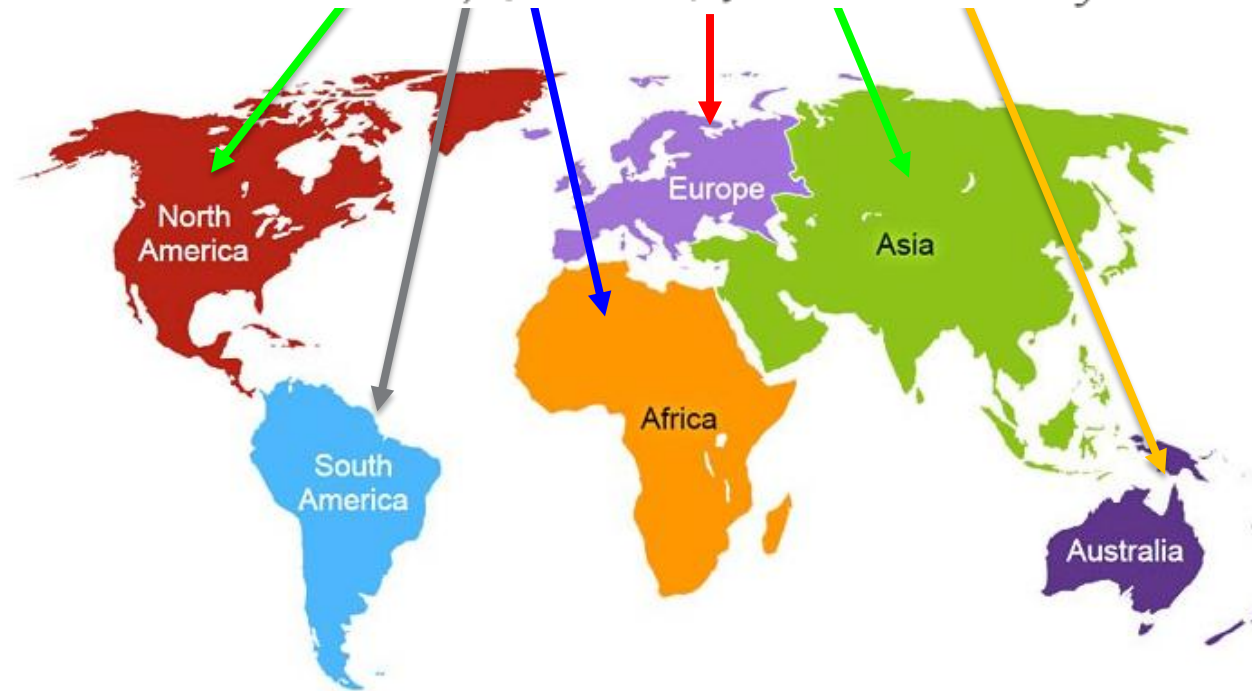





H2K Infosys, LLC
Dream, Strive & Achieve Victory



H2K Infosys is  business based in Atlanta, Georgia – United States
Providing Online IT training services world wide.

www.H2KINFOSYS.com

USA - +1-(770)-777-1269, UK - (020) 3371 7615
Training@H2KInfosys.com / H2KInfosys@Gmail.com

H2K INFOSYS PROVIDES WORLD CLASS SERVICES IN

**IT Trainings
with Real time
Project Work
for
Corporates
&
Individuals**

**Special IT
Training
for MS
Students in
US**

**Software Design,
Development,
QA – Manual,
Automation &
Performance
Testing,
Maintenance.**

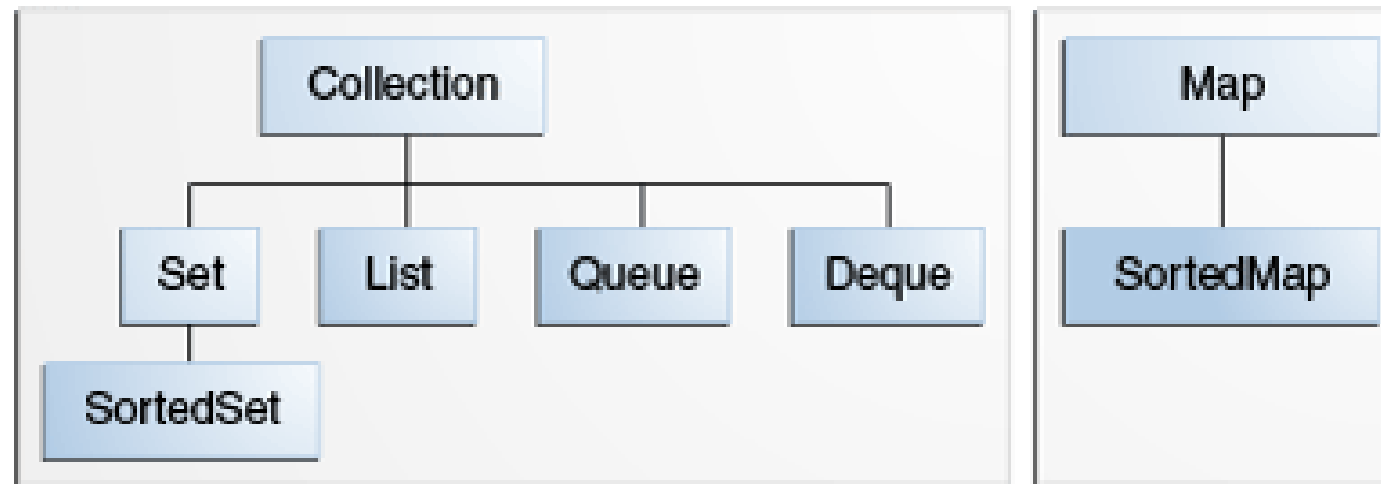
**IT Staff
Augmentation

Job Placement
Assistance

Tech Support**

Collections in Java.

- A collection is simply an object that groups multiple elements into a single unit. Collections are used to store, retrieve, manipulate, and communicate aggregate data.
- A collections framework is a unified architecture for representing and manipulating collections.
- The core collection interfaces encapsulate different types of collections, which are shown in the figure below.



The following list describes the core collection interfaces:

Collection	Description
Collection	the root of the collection hierarchy
Set	a collection that cannot contain duplicate elements
List	an ordered collection (sometimes called a <i>sequence</i>). Lists can contain duplicate elements
Queue	a collection used to hold multiple elements prior to processing. Queues typically, but do not necessarily, order elements in a FIFO (first-in, first-out) manner.
Deque	A linear collection that supports element insertion and removal at both ends.
Map	an object that maps keys to values (Map is not under Collection hierarchy)
SortedSet	a Set that maintains its elements in ascending order
SortedMap	a Map that maintains its mappings in ascending key order

Collection Interface Common Operations

Common methods applicable to all Collections.

Method	Action
add()	Adds an object to the collection
clear()	Removes all objects from the collection
contains()	Returns true if a specified object is an element within the collection
isEmpty()	Returns true if the collection has no elements
iterator()	Returns an Iterator object for the collection which may be used to retrieve an object
remove()	Removes a specified object from the collection
size()	Returns the number of elements in the collection

Collection Interface Bulk Operations

Bulk operations perform an operation on an entire Collection.

Method	Action
containsAll()	returns true if the target Collection contains all of the elements in the specified Collection.
addAll()	adds all of the elements in the specified Collection to the target Collection.
removeAll()	removes from the target Collection all of its elements that are also contained in the specified Collection.
retainAll()	removes from the target Collection all its elements that are not also contained in the specified Collection. That is, it retains only those elements in the target Collection that are also contained in the specified Collection.
clear()	removes all elements from the Collection.

The Set Interface

A Set is a Collection that cannot contain duplicate elements. It models the mathematical set abstraction. The Set interface contains only methods inherited from Collection and adds the restriction that duplicate elements are prohibited. .

Set	Details
HashSet	which stores its elements in a hash table, is the best-performing implementation; however it makes no guarantees concerning the order of iteration
TreeSet	which stores its elements in a red-black tree, orders its elements based on their values; it is substantially slower than HashSet
LinkedHashSet	which is implemented as a hash table with a linked list running through it, orders its elements based on the order in which they were inserted into the set (insertion-order)

The List Interface

The List interface includes operations for the following:

Positional access — manipulates elements based on their numerical position in the list

Search — searches for a specified object in the list and returns its numerical position

Iteration — extends Iterator semantics to take advantage of the list's sequential nature

Range-view — performs arbitrary range operations on the list.

List	Details
LinkedList	allows for constant-time insertions or removals, but only sequential access of elements
ArrayList	allow random access, so you can grab any element in constant time

The Map Interface

The Map interface maps unique keys to values. A key is an object that you use to retrieve a value at a later date. Given a key and a value, you can store the value in a Map object. After the value is stored, you can retrieve it by using its key.

Method	Action
void clear()	Removes all key/value pairs from the invoking map.
boolean containsKey(k)	Returns true if the invoking map contains k as a key. Otherwise, returns false.
boolean containsValue(v)	Returns true if the map contains v as a value. Otherwise, returns false.
Set entrySet()	Returns a Set that contains the entries in the map. The set contains objects of type Map.Entry. This method provides a set-view of the invoking map.
Object get(k)	Returns the value associated with the key k.
boolean isEmpty()	Returns true if the invoking map is empty. Otherwise, returns false.
Set keySet()	Returns a Set that contains the keys in the invoking map. This method provides a set-view of the keys in the invoking map.

The Map Interface

The HashMap class uses a hash table to implement the Map interface. This allows the execution time of basic operations, such as `get()` and `put()`, to remain constant even for large sets.

The LinkedHashMap extends HashMap and maintains a linked list of the entries in the map, in the order in which they were inserted.

Method	Action
<code>Object put(Object k, Object v)</code>	Puts an entry in the invoking map, overwriting any previous value associated with the key. The key and value are k and v, respectively. Returns null if the key did not already exist. Otherwise, the previous value linked to the key is returned.
<code>void putAll(Map m)</code>	Puts all the entries from m into this map.
<code>Object remove(Object k)</code>	Removes the entry whose key equals k.
<code>int size()</code>	Returns the number of key/value pairs in the map.
<code>Collection values()</code>	Returns a collection containing the values in the map. This method provides a collection-view of the values in the map.

A collection designed for holding elements prior to processing. Besides basic Collection operations, queues provide additional insertion, extraction, and inspection operations. .

Method	Action
E element()	Retrieves, but does not remove, the head of this queue.
boolean offer(E o)	Inserts the specified element into this queue, if possible.
E peek()	Retrieves, but does not remove, the head of this queue, returning null if this queue is empty.
E poll()	Retrieves and removes the head of this queue, or null if this queue is empty.
E remove()	Retrieves and removes the head of this queue.

Thank you

Your feedback is highly important to improve our course material and teaching methodologies. Please email your suggestions.

**USA +1-(770)-777-1269
Training@H2KInfosys.com**

**UK (020) 3371 7615
H2KInfosys@Gmail.com**

H2K Infosys is e-Verified business based in Atlanta, Georgia – United States

H2K Infosys acknowledges the proprietary rights of the trademarks and product names of other companies mentioned in this document.



H2K Infosys, LLC (hereinafter “H2K”) acknowledges the proprietary rights of the trademarks and product names of other companies mentioned in any of the training material including but not limited to the handouts, written material, videos, power point presentations, etc. All such training materials are provided to H2K students for learning purposes only. H2K students shall not use such materials for their private gain nor can they sell any such materials to a third party. Some of the examples provided in any such training materials may not be owned by H2K and as such H2K does not claim any proprietary rights for the same. H2K does not guarantee nor is it responsible for such products and projects. H2K acknowledges that any such information or product that has been lawfully received from any third party source is free from restriction and without any breach or violation of law whatsoever.