Java - Collections, Maps and Iterators

Collections in Java

- A collection is a data structure for holding elements
- java.util.Collection<T> is an interface implemented by many classes in Java
  - ArrayList<T>
  - Vector<T>
  - HashSet<T>
  - LinkedList<T>
  - etc. see the API

- Some methods in the Collection interface:
  - isEmpty(), contains(e), add(e), remove(e), iterator()

Maps in Java

- A map is an object that associates keys with values.
- A map cannot contain duplicate keys; each key can map to at most one value.
- java.util.Map<K,V> is an interface implemented by many classes in Java
  - HashMap<K,V>, Hashtable<K,V>
  - TreeMap<K,V>

- Some methods in the Map interface:
  - isEmpty(), containsKey(e), put(k,v), get(k), remove(k)
  - values(): Collection<V>

Iterators in Java

- An iterator is an object that cycles through all the elements in a collection.
- java.util.Iterator<T> is an interface with the following methods:
  - public T next() returns the next element in the collection (and advances)
  - public boolean hasNext() returns true if next() is not done.
  - public void remove() (Optional) removes the last element returned by next.
- You can get Iterators from Collections (and Maps):
  - ArrayList<Double> x = new ArrayList<Double>;
    Iterator<Double> it = x.iterator();
  - HashMap<String,Double> hm = new HashMap<String,Double>;
    Iterator<Double> it = hm.values().iterator();
import java.util.HashSet;
import java.util.Iterator;

public class HashSetIteratorDemo {
    public static void main(String args[]) {
        HashSet<String> s = new HashSet<String>();
        s.add("health");
        s.add("love");
        s.add("money");

        System.out.println("The set contains:");
        Iterator<String> i = s.iterator();
        while (i.hasNext())
            System.out.println(i.next());
    }
}

output:
The set contains:
love
money
health