**Java Collections & Util Package**

**List O**

List Of Collection Classes: **Sample Programs:**

* [Java Iterator Examples](http://www.java2novice.com/java-collections-and-util/iterator/)
* [Java ListIterator Examples](http://www.java2novice.com/java-collections-and-util/listiterator-example/)
* [Enumeration Examples](http://www.java2novice.com/java-collections-and-util/enumeration-example/)
* [Java Vector Examples](http://www.java2novice.com/java-collections-and-util/vector/)
* [Java ArrayList Examples](http://www.java2novice.com/java-collections-and-util/arraylist/)
* [Java LinkedList Examples](http://www.java2novice.com/java-collections-and-util/linkedlist/)
* [Java Hashtable Examples](http://www.java2novice.com/java-collections-and-util/hashtable/)
* [Java HashSet Examples](http://www.java2novice.com/java-collections-and-util/hashset/)
* [Java LinkedHashSet Examples](http://www.java2novice.com/java-collections-and-util/linked-hashset/)
* [Java TreeSet Examples](http://www.java2novice.com/java-collections-and-util/treeset/)
* [Java HashMap Examples](http://www.java2novice.com/java-collections-and-util/hashmap/)
* [Java TreeMap Examples](http://www.java2novice.com/java-collections-and-util/treemap/)
* [Java LinkedHashMap Examples](http://www.java2novice.com/java-collections-and-util/linkedhashmap/)
* [Java Collections Class Examples](http://www.java2novice.com/java-collections-and-util/collections/)

**Samp**List Of Util Classes Sample Programs:**le Programs:**

* [StringTokenizer Examples](http://www.java2novice.com/stringtokenizer/)
* [Properties Examples](http://www.java2novice.com/java-collections-and-util/properties/)
* [Java Date Examples](http://www.java2novice.com/java-util-date-example/)
* [Java Random Class Examples](http://www.java2novice.com/java-collections-and-util/random/)
* [Java Regular Expression Examples](http://www.java2novice.com/java-collections-and-util/regex/)
* [Java Zip Examples](http://www.java2novice.com/java-collections-and-util/zip/)

Learn how to play with collections in Java programming. Here are most commonly used examples:

1. [How to convert an array into a collection?](http://www.tutorialspoint.com/javaexamples/collection_array.htm)

## Solution:

Following example demonstrates to convert an array into a collection Arrays.asList(name) method of Java Util class.

import java.util.\*;

import java.io.\*;

public class ArrayToCollection{

public static void main(String args[])

throws IOException{

BufferedReader in = new BufferedReader

(new InputStreamReader(System.in));

System.out.println("How many elements

you want to add to the array: ");

int n = Integer.parseInt(in.readLine());

String[] name = new String[n];

for(int i = 0; i < n; i++){

name[i] = in.readLine();

}

**List list = Arrays.asList(name);**

System.out.println();

for(String li: list){

String str = li;

System.out.print(str + " ");

}

}

}

## Result:

The above code sample will produce the following result.

How many elements you want to add to the array:

red white green

red white green

1. [How to compare elements in a collection?](http://www.tutorialspoint.com/javaexamples/collection_compare.htm)

## Solution:

Following example compares the element of a collection by converting a string into a treeset using Collection.min() and Collection.max() methods of Collection class.

import java.util.Collections;

import java.util.Set;

import java.util.TreeSet;

class MainClass {

public static void main(String[] args) {

String[] coins = { "Penny", "nickel", "dime",

"Quarter", "dollar" };

Set set = new TreeSet();

for (int i = 0; i < coins.length; i++)

set.add(coins[i]);

System.out.println(Collections.min(set));

System.out.println(Collections.min(set,

String.CASE\_INSENSITIVE\_ORDER));

for(int i=0;i< =10;i++)

System.out.print(-);

System.out.println(Collections.max(set));

System.out.println(Collections.max(set,

String.CASE\_INSENSITIVE\_ORDER));

}

}

## Result:

The above code sample will produce the following result.

Penny

dime

----------

nickle

Quarter

1. [How to convert a collection into an array?](http://www.tutorialspoint.com/javaexamples/collection_conversion.htm)

## Solution:

Following example shows how to convert a collection to an array by using list.add() and list.toArray() method of Java Util class.

import java.util.\*;

public class CollectionToArray{

public static void main(String[] args){

List<String> list = new ArrayList<String>();

list.add("This ");

list.add("is ");

list.add("a ");

list.add("good ");

list.add("program.");

**String[] s1 = list.toArray(new String[0]);**

for(int i = 0; i < s1.length; ++i){

String contents = s1[i];

System.out.print(contents);

}

}

}

## Result:

The above code sample will produce the following result.

This is a good program.

1. [How to print a collection?](http://www.tutorialspoint.com/javaexamples/collection_print.htm)

## Solution:

Following example how to print a collection by using tMap.keySet(),tMap.values() and tMap.firstKey() methods of Java Util class .

import java.util.\*;

public class TreeExample{

public static void main(String[] args) {

System.out.println("Tree Map Example!\n");

TreeMap tMap = new TreeMap();

tMap.put(1, "Sunday");

tMap.put(2, "Monday");

tMap.put(3, "Tuesday");

tMap.put(4, "Wednesday");

tMap.put(5, "Thursday");

tMap.put(6, "Friday");

tMap.put(7, "Saturday");

System.out.println("Keys of tree map: "

+ tMap.keySet());

System.out.println("Values of tree map: "

+ tMap.values());

System.out.println("Key: 5 value: " + tMap.get(5)+ "\n");

System.out.println("First key: " + tMap.firstKey()

+ " Value: "

+ tMap.get(tMap.firstKey()) + "\n");

System.out.println("Last key: " + tMap.lastKey()

+ " Value: "+ tMap.get(tMap.lastKey()) + "\n");

System.out.println("Removing first data: "

+ tMap.remove(tMap.firstKey()));

System.out.println("Now the tree map Keys: "

+ tMap.keySet());

System.out.println("Now the tree map contain: "

+ tMap.values() + "\n");

System.out.println("Removing last data: "

+ tMap.remove(tMap.lastKey()));

System.out.println("Now the tree map Keys: "

+ tMap.keySet());

System.out.println("Now the tree map contain: "

+ tMap.values());

}

}

## Result:

The above code sample will produce the following result.

C:\collection>javac TreeExample.java

C:\collection>java TreeExample

Tree Map Example!

Keys of tree map: [1, 2, 3, 4, 5, 6, 7]

Values of tree map: [Sunday, Monday, Tuesday, Wednesday,

Thursday, Friday, Saturday]

Key: 5 value: Thursday

First key: 1 Value: Sunday

Last key: 7 Value: Saturday

Removing first data: Sunday

Now the tree map Keys: [2, 3, 4, 5, 6, 7]

Now the tree map contain: [Monday, Tuesday, Wednesday,

Thursday, Friday, Saturday]

Removing last data: Saturday

Now the tree map Keys: [2, 3, 4, 5, 6]

Now the tree map contain: [Monday, Tuesday, Wednesday,

Thursday, Friday]

1. [How to make a collection read-only?](http://www.tutorialspoint.com/javaexamples/collection_readonly.htm)

How to make a collection read-only ?

## Solution:

Following example shows how to make a collection read-only by using Collections.unmodifiableList() method of Collection class.

import java.util.ArrayList;

import java.util.Arrays;

import java.util.Collections;

import java.util.HashMap;

import java.util.HashSet;

import java.util.List;

import java.util.Map;

import java.util.Set;

public class Main {

public static void main(String[] argv)

throws Exception {

List stuff = Arrays.asList(new String[] { "a", "b" });

List list = new ArrayList(stuff);

list = Collections.unmodifiableList(list);

try {

list.set(0, "new value");

}

catch (UnsupportedOperationException e) {

}

Set set = new HashSet(stuff);

set = Collections.unmodifiableSet(set);

Map map = new HashMap();

map = Collections.unmodifiableMap(map);

System.out.println("Collection is read-only now.");

}

}

## Result:

The above code sample will produce the following result.

Collection is read-only now.

1. [How to remove a specific element from a collection?](http://www.tutorialspoint.com/javaexamples/collection_remove.htm)

## Solution:

Following example demonstrates how to remove a certain element from a collection with the help of collection.remove() method of Collection class.

import java.util.\*;

public class CollectionTest {

public static void main(String [] args) {

System.out.println( "Collection Example!\n" );

int size;

HashSet collection = new HashSet ();

String str1 = "Yellow", str2 = "White", str3 =

"Green", str4 = "Blue";

Iterator iterator;

collection.add(str1);

collection.add(str2);

collection.add(str3);

collection.add(str4);

System.out.print("Collection data: ");

iterator = collection.iterator();

while (iterator.hasNext()){

System.out.print(iterator.next() + " ");

}

System.out.println();

collection.remove(str2);

System.out.println("After removing [" + str2 + "]\n");

System.out.print("Now collection data: ");

iterator = collection.iterator();

while (iterator.hasNext()){

System.out.print(iterator.next() + " ");

}

System.out.println();

size = collection.size();

System.out.println("Collection size: " + size + "\n");

}

}

## Result:

The above code sample will produce the following result.

Collection Example!

Collection data: Blue White Green Yellow

After removing [White]

Now collection data: Blue Green Yellow

Collection size: 3

1. [How to reverse a collection?](http://www.tutorialspoint.com/javaexamples/collection_reverse.htm)

## Solution:

Following example demonstratres how to reverse a collection with the help of listIterator() and Collection.reverse() methods of Collection and Listiterator class .

import java.util.ArrayList;

import java.util.Collections;

import java.util.List;

import java.util.ListIterator;

class UtilDemo3 {

public static void main(String[] args) {

String[] coins = { "A", "B", "C", "D", "E" };

List l = new ArrayList();

for (int i = 0; i < coins.length; i++)

l.add(coins[i]);

ListIterator liter = l.listIterator();

System.out.println("Before reversal");

while (liter.hasNext())

System.out.println(liter.next());

Collections.reverse(l);

liter = l.listIterator();

System.out.println("After reversal");

while (liter.hasNext())

System.out.println(liter.next());

}

}

## Result:

The above code sample will produce the following result.

Before reversal

A

B

C

D

E

After reversal

E

D

C

B

A

1. [How to shuffle the elements of a collection?](http://www.tutorialspoint.com/javaexamples/collection_shuffle.htm)

## Solution:

Following example how to shuffle the elements of a collection with the help of Collections.shuffle() method of Collections class.

import java.util.ArrayList;

import java.util.Collections;

import java.util.List;

public class Main {

public static void main(String[] argv)

throws Exception {

String[] alpha = { "A", "E", "I", "O", "U" };

List list = new ArrayList(alpha);

Collections.shuffle(list);

System.out.println("list");

}

}

## Result:

The above code sample will produce the following result.

I

O

A

U

E

1. [How to get the size of a collection?](http://www.tutorialspoint.com/javaexamples/collection_size.htm)

## Solution:

Following example shows how to get the size of a collection by the use of collection.add() to add new data and collection.size() to get the size of the collection of Collections class.

import java.util.\*;

public class CollectionTest {

public static void main(String [] args) {

System.out.println( "Collection Example!\n" );

int size;

HashSet collection = new HashSet ();

String str1 = "Yellow", str2 = "White", str3 =

"Green", str4 = "Blue";

Iterator iterator;

collection.add(str1);

collection.add(str2);

collection.add(str3);

collection.add(str4);

System.out.print("Collection data: ");

iterator = collection.iterator();

while (iterator.hasNext()){

System.out.print(iterator.next() + " ");

}

System.out.println();

size = collection.size();

if (collection.isEmpty()){

System.out.println("Collection is empty");

}

else{

System.out.println( "Collection size: " + size);

}

System.out.println();

}

}

## Result:

The above code sample will produce the following result.

Collection Example!

Collection data: Blue White Green Yellow

Collection size: 4

1. [How to iterate through elements of HashMap?](http://www.tutorialspoint.com/javaexamples/collection_iterate.htm)

## Solution:

Following example uses iterator Method of Collection class to iterate through the HashMap.

import java.util.\*;

public class Main {

public static void main(String[] args) {

HashMap< String, String> hMap =

new HashMap< String, String>();

hMap.put("1", "1st");

hMap.put("2", "2nd");

hMap.put("3", "3rd");

Collection cl = hMap.values();

Iterator itr = cl.iterator();

while (itr.hasNext()) {

System.out.println(itr.next());

}

}

}

## Result:

The above code sample will produce the following result.

3rd

2nd

1st

1. [How to use enumeration to display contents of HashTable?](http://www.tutorialspoint.com/javaexamples/collection_enumeration.htm)

## Solution:

Following example uses hasMoreElements & nestElement Methods of Enumeration Class to display the contents of the HashTable.

import java.util.Enumeration;

import java.util.Hashtable;

public class Main {

public static void main(String[] args) {

Hashtable ht = new Hashtable();

ht.put("1", "One");

ht.put("2", "Two");

ht.put("3", "Three");

Enumeration e = ht.elements();

while(e.hasMoreElements()){

System.out.println(e.nextElement());

}

}

}

## Result:

The above code sample will produce the following result.

Three

Two

One

1. [How to get Set view of Keys from Java Hashtable?](http://www.tutorialspoint.com/javaexamples/collection_hashtable_key.htm)

How to set view of Keys from Java Hashtable?

## Solution:

Following example uses keys() method to get Enumeration of Keys of the Hashtable.

import java.util.Enumeration;

import java.util.Hashtable;

public class Main {

public static void main(String[] args) {

Hashtable ht = new Hashtable();

ht.put("1", "One");

ht.put("2", "Two");

ht.put("3", "Three");

Enumeration e = ht.keys();

while (e.hasMoreElements()){

System.out.println(e.nextElement());

}

}

}

## Result:

The above code sample will produce the following result.

3

2

1

1. [How to find min & max of a List?](http://www.tutorialspoint.com/javaexamples/collection_minmax.htm)

## Solution:

Following example uses min & max Methods to find minimum & maximum of the List.

import java.util.\*;

public class Main {

public static void main(String[] args) {

List list = Arrays.asList("one Two three Four five six

one three Four".split(" "));

System.out.println(list);

System.out.println("max: " + Collections.max(list));

System.out.println("min: " + Collections.min(list));

}

}

## Result:

The above code sample will produce the following result.

[one, Two, three, Four, five, six, one, three, Four]

max: three

min: Four

1. [How to find a sublist in a List?](http://www.tutorialspoint.com/javaexamples/collection_sublist.htm)

## Solution:

Following example uses indexOfSubList() & lastIndexOfSubList() to check whether the sublist is there in the list or not & to find the last occurance of the sublist in the list.

import java.util.\*;

public class Main {

public static void main(String[] args) {

List list = Arrays.asList("one Two three Four five

six one three Four".split(" "));

System.out.println("List :"+list);

List sublist = Arrays.asList("three Four".split(" "));

System.out.println("SubList :"+sublist);

System.out.println("indexOfSubList: "

+ Collections.indexOfSubList(list, sublist));

System.out.println("lastIndexOfSubList: "

+ Collections.lastIndexOfSubList(list, sublist));

}

}

## Result:

The above code sample will produce the following result.

List :[one, Two, three, Four, five, six, one, three, Four]

SubList :[three, Four]

indexOfSubList: 2

lastIndexOfSubList: 7

1. [How to replace an element in a list?](http://www.tutorialspoint.com/javaexamples/collection_replace.htm)

## Solution:

Following example uses replaceAll() method to replace all the occurance of an element with a different element in a list.

import java.util.\*;

public class Main {

public static void main(String[] args) {

List list = Arrays.asList("one Two three Four five six

one three Four".split(" "));

System.out.println("List :"+list);

Collections.replaceAll(list, "one", "hundread");

System.out.println("replaceAll: " + list);

}

}

## Result:

The above code sample will produce the following result.

List :[one, Two, three, Four, five, six, one, three, Four]

replaceAll: [hundread, Two, three, Four, five, six,

hundread, three, Four]

1. [How to rotate elements of the List?](http://www.tutorialspoint.com/javaexamples/collection_rotate.htm)

## Solution:

Following example uses rotate() method to rotate elements of the list depending on the 2nd argument of the method.

import java.util.\*;

public class Main {

public static void main(String[] args) {

List list = Arrays.asList("one Two three Four five

six".split(" "));

System.out.println("List :"+list);

Collections.rotate(list, 3);

System.out.println("rotate: " + list);

}

}

## Result:

The above code sample will produce the following result.

List :[one, Two, three, Four, five, six]

rotate: [Four, five, six, one, Two, three]