**Java Collections Utility Class**

[**<<Previous**](http://www.w3resource.com/java-tutorial/java-maps.php)[**Next>>**](http://www.w3resource.com/java-tutorial/java-defining-instantiating-and-starting-threads.php)

**Introduction**

The Collections utility class consists exclusively of static methods that operate on or return collections. It contains polymorphic algorithms that operate on collections, "wrappers", which return a new collection backed by a specified collection,

Some useful method in Collections class:

|  |  |
| --- | --- |
| Method Signature | Description |
| Collections.sort(List myList) | Sort the myList (implementation of any List interface) provided in argument in natural ordering. |
| Collections.sort(List, comparator c) | Sort the myList(implementation of any List interface) as per comparator c ordering (c class should implement comparator interface) |
| Collections.shuffle(List myList) | Puts the elements of myList ((implementation of any List interface)in random order |
| Collections.reverse(List myList) | Reverses the elements of myList ((implementation of any List interface) |
| Collections.binarySearch(List mlist, T key) | Searches the mlist (implementation of any List interface) for the specified object using the binary search algorithm. |
| Collections.copy(List dest, List src) | Copy the source List into dest List. |
| Collections.frequency(Collection c, Object o) | Returns the number of elements in the specified collection class c (which implements Collection interface can be List, Set or Queue) equal to the specified object |
| Collections.synchronizedCollection(Collection c) | Returns a synchronized (thread-safe) collection backed by the specified collection. |

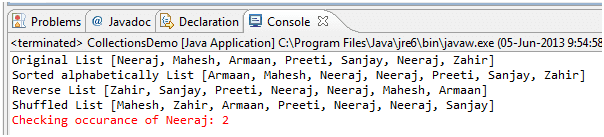
Lets take example of List sorting using Collection class. We can sort any Collection using “Collections” utility class. i.e.; ArrayList of Strings can be sorted alphabetically using this utility class. ArrayList class itself is not providing any methods to sort. We use Collections class static methods to do this. Below program shows use of reverse(), shuffle(), frequency() methods as well.

**Java Code**

[view plaincopy to clipboardprint?](http://www.w3resource.com/java-tutorial/java-collections-utility-class.php)

1. **package** utility;
3. **import** java.util.Collections;
4. **import** java.util.ArrayList;
5. **import** java.util.List;
7. **public** **class** CollectionsDemo {
9. **public** **static** **void** main(String[] args) {
10. List<String>student<String>List = **new** ArrayList();
11. studentList.add("Neeraj");
12. studentList.add("Mahesh");
13. studentList.add("Armaan");
14. studentList.add("Preeti");
15. studentList.add("Sanjay");
16. studentList.add("Neeraj");
17. studentList.add("Zahir");
19. System.out.println("Original List " + studentList);
21. Collections.sort(studentList);
22. System.out.println("Sorted alphabetically List " + studentList);
24. Collections.reverse(studentList);
25. System.out.println("Reverse List " + studentList);
26. Collections.shuffle(studentList);
27. System.out.println("Shuffled List " + studentList);
28. System.out.println("Checking occurance of Neeraj: "
29. + Collections.frequency(studentList, "Neeraj"));
30. }
31. }

**Output**



Using Collections class we can copy one type of collection to other type. Collections provide us copy method to copy all the elements from source to destination. Below program demonstrate the use of copy function. Here size of source collection and destination collection should be same else we will get following exception.

java collection uitlity image2

**Java Code**

[view plaincopy to clipboardprint?](http://www.w3resource.com/java-tutorial/java-collections-utility-class.php)

1. **package** utility;
3. **import** java.util.Collections;
4. **import** java.util.\*;
6. **public** **class** CopyListDemo {
8. **public** **static** **void** main(String[] args) {
9. List <Integer>myFirstList = **new** ArrayList<Integer>();
10. List <Integer> mySecondList = **new** ArrayList<Integer>();
11. myFirstList.add(10);
12. myFirstList.add(20);
13. myFirstList.add(20);
14. myFirstList.add(50);
15. myFirstList.add(70);

18. mySecondList.add(11);
19. mySecondList.add(120);
20. mySecondList.add(120);
21. mySecondList.add(150);
22. mySecondList.add(170);
24. System.out.println("First List-"+ myFirstList);
25. System.out.println("Second List-"+ mySecondList);
26. Collections.copy(mySecondList, myFirstList );
27. System.out.println("Second List After Copy-"+ mySecondList);
28. }
29. }

**Output**

