Convert List to array.

In previous post we saw some Array questions like finding min, finding max, summation, average, reverse, linear search, Fibonacci, last index of element and shifting elements to left or right. Navigate to Arrays link for more questions.

In this post we will see how to convert [List<E>](http://data-structure-learning.blogspot.com/2015/05/java-collections-part-5list-interface.html) to array.

**First**, let us say that List<E> consist of Object, more specifically String objects. So we can directly convert that List<E> to String[]. We use

**<T> T[] toArray(T[] a); method of List<E> interface.**

Below is the code

/\*\*

\* This method is used to convert the List to String[]

\* we use <T> T[] toArray(T[] a); method of List<E> interface.

\* \*/

**public** **static** String[] listToArray(List<String> list) {

String[] str = list.toArray(**new** String[list.size()]);

**return** str;

}

**Second**, let’s assume that we have a List<Integer> that we want to convert to int[]. Now we know that we can use intValue() of Integer class to get int value from Integer object. If you don’t know about it then remember that intValue() method returns the int value of Integer object. Also Java 5 and older versions have support of Autoboxing. Below is the code:

/\*\*

\* This method demonstrate the conversion of

\* List<Integer> to int[].

\*

\* arr[i] = ints.get(i); is example of autoboxing.

\* \*/

**public** **static** **int**[] listToPrimitiveArray(List<Integer> ints) {

**int**[] arr = **new** **int**[ints.size()];

**for** (**int** i = 0; i < ints.size(); i++) {

arr[i] = ints.get(i);

//Below line can also be used.

//arr[i] = ints.get(i).intValue();

}

**return** arr;

}

**Third**, We can use the stream() method of [Collection<E>](http://data-structure-learning.blogspot.com/2015/05/java-collections-part-4collection.html) interface(introduced in Java 8) then map the stream output to int using intValue() method and convert them to array. Below is the code:

/\*\*

\* Below method uses Java 8 for conversion of List<Integer> to int[].

\*

\* Integer::intValue is new syntax which means

\* Integer is class and intValue is instance method of class Integer.

\* \*/

**public** **static** **int**[] listToPrimitiveArray8(List<Integer> ints) {

**int**[] arr = ints.stream().mapToInt(Integer::intValue).toArray();

**return** arr;

}

**Fourth** method is using external library called Google Guava. We can use Ints class static method toArray() to convert List<Integer> to int[].

/\*\*

\* This method uses Ints class of Google Guava for conversion of

\* List<Integer> to int[]

\*\*/

**public** **static** **int**[] listToPrmitiveArrayGoogleGuava(List<Integer> ints){

**int**[] arr = Ints.*toArray*(ints);

**return** arr;

}