**Java ArrayList and Vector**

**Introduction**

In addition to the Arrays class, Java provides an ArrayList class which can be used to create containers that store lists of objects. ArrayList can be considered as a growable array. It gives you fast iteration and fast random access. ArrayList implements the new RandomAccess interface—a marker interface (meaning it has no methods) that says, "This list supports fast (generally constant time) random access." Choose this over a LinkedList when you need fast iteration but aren't as likely to be doing a lot of insertion and deletion.

Earlier versions of Java have one legacy collection class called Vector which is very much similar to ArrayList. Vector implements a dynamic array. A Vector is basically the same as an ArrayList, but Vector methods are synchronized for thread safety. You'll normally want to use ArrayList instead of Vector because the synchronized methods add a performance hit you might not need. In this tutorial we will discuss about ArrayList only considering all is applicable to Vector as well.

The java.util.ArrayList class is one of the most commonly used of all the classes in the Collections Framework.An ArrayList is dynamically resizable, meaning that its size can change during program execution. This means that:

* You can add an item at any point in an ArrayList container and the array size expands automatically to accommodate the new item.
* You can remove an item at any point in an ArrayList container and the array size contracts automatically.

To state the obvious: Arraylist is an ordered collection (by index), but not sorted.

To use the ArrayList class, you must use the following import statement:

import java.util.ArrayList;

Then, to declare an ArrayList, you can use the default constructor, as in the following example:

ArrayList names = new ArrayList();

The default constructor creates an ArrayList with a capacity of 10 items. The capacity of an ArrayList is the number of items it can hold without having to increase its size. Other constructors of ArrayList as follows,

ArrayList names = new ArrayList(int size);

ArrayList names = new ArrayList(Collection c);

You can also specify a capacity/size of initial ArrayList as well as create ArrayList from other collection types.

Some of the advantages ArrayList has over arrays are

* It can grow dynamically.
* It provides more powerful insertion and search mechanisms than arrays.

**ArrayList methods**

|  |  |
| --- | --- |
| **Method** | **Purpose** |
| public void add(Object) public void add(int, Object) | Adds an item to an ArrayList. The default version adds an item at the next available location; an overloaded version allows you to specify a position at which to add the item |
| public void remove(int) | Removes an item from an ArrayList at a specified location |
| public void set(int, Object) | Alters an item at a specified ArrayList location |
| Object get(int) | Retrieves an item from a specified location in an ArrayList |
| public int size() | Returns the current ArrayList size |
|  |  |

Java Program to demonstrate use of all above methods described above. Here we are creating ArrayList named myList and adding objects using add() method as well as using index based add method, then printing all the objects using for loop. Then there we demonstrate use of get(), contains(), and size() methods. Output of program is shown below the java code.

**Java Code**

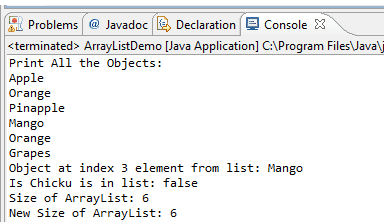
[view plaincopy to clipboardprint?](http://www.w3resource.com/java-tutorial/java-arraylist-and-vector.php)

1. **package** arraylist;
3. **import** java.util.ArrayList;
5. **public** **class** ArrayListDemo {


9. **public** **static** **void** main(String[] args) {
11. //declaring Arraylist of String objects
13. ArrayList<String> myList = **new** ArrayList<String>();
15. //Adding objects to Array List at default index
17. myList.add("Apple");
19. myList.add("Mango");
21. myList.add("Orange");
23. myList.add("Grapes");
25. //Adding object at specific index
27. myList.add(1, "Orange");
29. myList.add(2,"Pinapple");


33. System.out.println("Print All the Objects:");
35. **for**(String s:myList){
37. System.out.println(s);
39. }
41. System.out.println("Object at index 3 element from list: "+ myList.get(3));
43. System.out.println("Is Chicku is in list: " + myList.contains("Chicku"));
45. System.out.println("Size of ArrayList: " + myList.size());
47. myList.remove("Papaya");
49. System.out.println("New Size of ArrayList: "+ myList.size());
51. }
53. }

**Output**



**Summary**

* ArrayList and Vector is similar classes only difference is Vector has all method synchronized. Both class simple terms can be considered as growable array.
* ArrayList should be used in application when we need to search objects from list based on index.
* ArrayList performance degrades when there is lots of insert and update operation in middle of the list.