Search for an Element in an ArrayList

[ArrayList<E>](http://data-structure-learning.blogspot.com/2015/05/java-collections-part-5list-interface.html) is a classed backed by an array which provides us several advantages over arrays.

Read here for [iterating ArrayList<E> in 6 different ways](http://data-structure-learning.blogspot.com/2015/05/java-collections-part-6iterating-over.html). There are also few [differences between ArrayList<E> and LinkedList<E>](http://data-structure-learning.blogspot.com/2015/05/difference-between-arraylist-and_28.html), few [similarities between ArrayList<E> and LinkedList<E>](http://data-structure-learning.blogspot.com/2015/05/similarities-between-arraylist-and.html) and [when to use either of them](http://data-structure-learning.blogspot.com/2015/05/when-to-use-arraylist-and-linkedlist.html).

Now let us search for element in ArrayList.

We will use contains method to search for element.

**boolean** contains(Object o);

contains(Object o) takes one parameter of type Object. This method does a linear scan in List and tries to search element. As linear scan is performed time taken to search element is O(n).

Let us see the code of contains(Object o) method of ArrayList class.

**public** **boolean** contains(Object o) {

**return** indexOf(o) >= 0;

}

So contains(Object o) internally uses indexOf(Object o) method.

**public** **int** indexOf(Object o) {

**if** (o == **null**) {

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

**if** (elementData[i]==**null**)

**return** i;

} **else** {

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

**if** (o.equals(elementData[i]))

**return** i;

}

**return** -1;

}

indexOf(Object o) is a method that searches for element in list. If found then index is returned else -1 is returned.

Now again look at contains(Object o) method. If the index returned by indexOf(Object o) is greater or equal to 0 then only it will return true else false.

So this is how contains(Object o) method works internally.

Below code demonstrates the contains(Object o) method.

**package** org.collections;

**import** java.util.ArrayList;

**import** java.util.List;

**public** **class** ArrayListContains {

**public** **static** **void** main(String[] args) {

List<String> list = *populateNames*();

String s1 = "Eddard";

**boolean** b1 = *containsMethodDemo*(list, s1);

System.***out***.println(s1 + " in list? " + b1);

String s2 = "John";

**boolean** b2 = *containsMethodDemo*(list, s2);

System.***out***.println(s2 + " in list? " + b2);

}

**public** **static** List<String> populateNames() {

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

list.add("Chris");

list.add("Tony");

list.add("Eddard");

list.add("Robert");

list.add("Thor");

**return** list;

}

**public** **static** **boolean** containsMethodDemo(List<String> list, String input) {

**if** (list == **null**) {

**throw** **new** NullPointerException("Input List must not be empty");

}

**return** list.contains(input);

}

}

Output:

Eddard in list? true

John in list? false