```
//Program to demonstrate ArrayList
package com.tnsif.collection.list;

import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.Iterator;
import java.util.List;
import java.util.ListIterator;

public class ArrayListDemo {

 public static void main(String[] args) {

  List list1 = new ArrayList();// creates empty raw ArrayList
(allows heterogeneous elements)
  System.out.println("Size : "+list1.size());
  System.out.println("Is list empty? "+list1.isEmpty());
  list1.add(10);
  list1.add(20);
  list1.add(true);
  list1.add(false);
  list1.add(20);
  list1.add("Hello");
  list1.add(56.78);
  list1.add(20);
  list1.add('A');
  list1.add(5, "Hi");
  list1.add(20);
  System.out.println("List is "+list1);

  System.out.println("Is list conatins 15?
"+list1.contains(15));

  list1.remove(false);
```

```
  System.out.println("List is "+list1);


 //   System.out.println(list1.remove(20)); //4 - index
System.out.println(list1);


  System.out.println("Element at 5 location is :
"+list1.get(5));

  System.out.println("Element removed :
"+list1.remove(list1.lastIndexOf(20))); // remove 20 (last
occurrence)
  System.out.println("Element removed :
"+list1.remove(list1.indexOf(20))); // remove 20 (first
occurrence)
  System.out.println("List is "+list1);

/*
 * Collections.sort(list1);//RTE - ClassCastException
 * System.out.println("List is "+list1);
 */

list1.clear();
System.out.println("List is "+list1);

// Using Generic we can create Homogeneous List
List<String> names = new ArrayList<String>();
names.add("Amit");
names.add("Sumit");
names.add("Ankit");
names.add("Rohit");
names.add("Likshit");
//names.add(10); //CTE - Generic are type safety
```

```
  System.out.println("Name List is "+names);

  Collections.reverse(names);
  System.out.println("Reverse Name List is "+names);

  System.out.println("Is \'Ankit\' contains in name list?
"+names.contains("Ankit"));

  System.out.println("Name List Before Sorting is "+names);
  Collections.sort(names);
  System.out.println("Sorting in Ascending order " + names);

  Collections.reverse(names);
  System.out.println("Sorting in Descending order " +
names);

  // Traversing a list
  System.out.println("---------- Traversing a list ----------");
  Iterator<String> i = names.iterator();
  while (i.hasNext()) {
   String nm = i.next();
   System.out.println(nm);
   if (nm.equals("Ankit"))
    i.remove();
  }

  System.out.println("Name list is "+names);
  System.out.println("---------- Traversing a list in
backward manner ----------");
  ListIterator<String> li=names.listIterator(names.size());
  while(li.hasPrevious())
  {
   String nm = li.previous();
   System.out.println(nm);
  }
```

```
    }

}
```