

JAVA REPORT

Date:	15/06/2020	Name:	SAFIYA BANU
Course:	JAVA	USN:	4AL16EC061
Topic:	<ol style="list-style-type: none">1. ArrayList: Arrays the Easy Way2. Linked Lists3. HashMap: Retrieving Objects via a Key4. Sorted Maps5. Sets6. Using Custom Objects in Sets and as Keys in Maps7. Sorting Lists	Semester & Section:	8 TH B
Github Repository:	Safiya-Courses		

Java Tutorial for Complete Beginners
★ Leave a rating
🔄 Your progress
🔗 Share

```

import java.util.*;

public class App {
    public static void main(String[] args) {
        // Adding elements
        ArrayList<Integer> numbers = new ArrayList<>();
        numbers.add(100);
        numbers.add(40);

        // Iterating over the list
        System.out.println("Iteration #1: ");
        for (int i = 0; i < numbers.size(); i++) {
            System.out.println(numbers.get(i));
        }

        // Removing elements
        numbers.remove(1);

        // This is VERY slow
        numbers.remove(0);
        System.out.println("Iteration #2: ");
        for (Integer value : numbers) {
            System.out.println(value);
        }
    }
}

```

Course content

- ☐ 53. ArrayList: Arrays the Easy Way
10min [Resources](#)
- ☐ 54. Linked Lists
14min [Resources](#)
- ☐ 55. HashMap: Retrieving Objects via a Key
10min [Resources](#)
- ☐ 56. Sorted Maps
12min [Resources](#)
- ☐ 57. Sets
16min [Resources](#)
- ☐ 58. Using Custom Objects in Sets and as Keys in Maps
11min [Resources](#)
- ☐ 59. Sorting Lists
21min [Resources](#)
- ☐ 60. Natural Ordering
20min [Resources](#)
- ☐ 61. Queues
18min [Resources](#)

About this course
 Learn to program using the Java programming language

Java Tutorial for Complete Beginners
🔄 Your progress
🔗 Share

```

import java.util.*;

public class App {
    public static void main(String[] args) {
        // ArrayList manage arrays internally.
        // [0][1][2][3][4][5] ...
        List<Integer> arrayList = new ArrayList<>();

        // LinkedList consists of elements where each element
        // has a reference to the previous and next element
        // [0] -> [1]
        List<Integer> linkedList = new LinkedList<>();

        doTimings("ArrayList", arrayList);
        doTimings("LinkedList", linkedList);

        private static void doTimings(String type, List<Integer> list) {
            for (int i = 0; i < 1000; i++) {
                // ...
            }
        }
    }
}

```

Course content

- ☒ 54. Linked Lists
14min [Resources](#)
- ☐ 55. HashMap: Retrieving Objects via a Key
10min [Resources](#)
- ☐ 56. Sorted Maps
12min [Resources](#)
- ☐ 57. Sets
16min [Resources](#)
- ☐ 58. Using Custom Objects in Sets and as Keys in Maps
11min [Resources](#)
- ☐ 59. Sorting Lists
21min [Resources](#)
- ☐ 60. Natural Ordering
20min [Resources](#)
- ☐ 61. Queues
18min [Resources](#)
- ☐ 62. Using Iterators
9min [Resources](#)

About this course
 Learn to program using the Java programming language

Array of ArrayList in Java

We often come across 2D arrays where most of the part in the array is empty. Since space is a huge problem, we try to optimize the space. One such solution is to use jagged array when we know the length of each row in the array, but the prob

specifically know the length of each of the rows. Here we use ArrayList since the
Following is a Java program to demonstrate the above concept.

LinkedList in Java

Linked List are linear data structures where the elements are not stored in contiguous locations and every element has a data part and address part. The elements are linked using pointers and addresses. Each element is known as a node. Due to ease of insertions and deletions, they are preferred over the arrays. It also has few disadvantages like the nodes are not contiguous, instead we need to start from the head and follow through the link to reach to a node. To store the elements in a linked list we use a doubly linked list which provides a linear data structure and also used to implement list and deque interfaces.

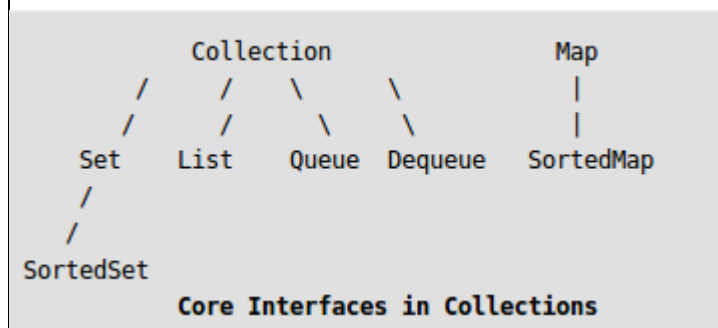
In Java, LinkedList class implements the list interface. The LinkedList class also consists of various constructors and methods for collections.

Constructors for Java LinkedList:

1. `LinkedList()`: Used to create an empty linked list.
2. `LinkedList(Collection C)`: Used to create a ordered list which contains all the elements of a specified collection using the collection's iterator.

SortedMap Interface in Java with Examples

SortedMap is an interface in collection framework. This interface extends Map interface and provides a total ordering on its keys (can be traversed in sorted order of keys). Example class that implements this interface is TreeMap.



The main characteristic of a SortedMap is that, it orders the keys by their natural ordering, or by a specified comparator. You can use a TreeMap when you want a map that satisfies the following criteria:

Collections.sort() in Java with Examples

`java.util.Collections.sort()` method is present in `java.util.Collections` class. It is used to sort the elements present in the array in ascending order.

It works similar to `java.util.Arrays.sort()` method but it is better then as it can sort the elements of Array as well as List more present in it.

```
public static void sort(List myList)
```

`myList` : A List type object we want to sort.

This method doesn't return anything