

List

import java.util list;

public class Main {

public static void main (String [] args) {

ArrayList <String> obj = new ArrayList <>();

obj.add("One");

obj.add("Two");

obj.add("Three");

System.out.println("ArrayList : " + obj);

output:- ["One", "Two", "Three"];

}

} ArrayListimport java.util. ~~list~~ ArrayList;

class Main {

public static void main (String [] args) {

List <Integer> numbers = new ArrayList <>();

numbers.add(1);

numbers.add(2);

numbers.add(3);

System.out.println("List: " + numbers);

int get number = numbers.get(2);

System.out.println("Accessed Element: " + get number);

int remove number = numbers.remove(1);

System.out.println("Remove element: " + ~~Remove~~ remove Number);

}

Linked list

```
import java.util.List;  
import java.util.LinkedList;
```

```
class Main {
```

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

```
        List<Integer> numbers = new LinkedList LinkedList<>();
```

```
        numbers.add(1);
```

```
        numbers.add(2);
```

```
        numbers.add(3);
```

```
        System.out.println("Accessed Element : " + numbers);
```

```
        int index = numbers.indexOf(2)
```

```
        System.out.println("Position of 2 " + index);
```

```
        int removed Number = numbers.remove(1);
```

```
        System.out.println("Position of 2 " + index);
```

```
        int removed Number = numbers.remove(1);
```

```
        System.out.println("Removed element " + removed number);
```

```
    }
```

```
}
```

Vector

```
import java.util.Vector;
```

```
class Main {
```

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

```
        Vector<String> fruits = new Vector<>();
```

```
        fruits.add("Apple");
```

```
        fruits.add("Orange");
```

```
        fruits.add("Mango");
```

```
        System.out.println("Vector : " + fruits);
```

```
        String element = fruits.get(2);
```



```

int size = fruits.size();
System.out.println("Size: " + size);
Iterators < String > iterate = fruits.iterator();
System.out.print("Vector");
while (iterate.hasNext()) {
    System.out.print(iterate.next());
}
}

```

Collection reverse

```

import java.util.*;
import java.util.Iterator;
import java.util.LinkedList;
import java.util.Collections;
class Main {
    public static void Main (String [] args) {
        List < String > fruits = new LinkedList < > ();
        fruits.add ("Apple");
        fruits.add ("Orange");
        fruits.add ("Mango");
        System.out.println ("Linked list: " + fruits);
        fruits.add (2, "Banana");
        System.out.println ("Linked list: " + fruits);
        Collections.reverse (fruits);
        System.out.println ("Fruits in reverse order: " + fruits);
        Collections.sort (fruits);
        System.out.println ("Fruits in Ascending order: " + fruits);
    }
}

```



```

System.out.println("Fruits in the basket : ");
for (int i = fruits.size() - 1; i >= 0; i--) {
    System.out.println(fruits.get(i));
}
}
}

```

Stack (push, Pop, peek, empty).

```

import java.lang.*;
import java.util.Stack;

```

Class Main {

public static void main (String[] args) {

Stack <String> fruits = new Stack<>();

fruits.push("Apple");

fruits.push("orange");

fruits.push("Mango");

System.out.println("Stack:" + fruits);

String remove = fruits.pop();

System.out.println("Stack:" + remove);

fruits.push("Pineapple");

System.out.println("Stack:" + fruits);

String display = fruits.peek();

System.out.println("Stack:" + display);

boolean e = fruits.empty();

System.out.print("Is the stack

Queue

```
import java.lang.*;  
import java.util.LinkedList;  
import java.util.Queue;
```

Class Main {

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

```
    Queue <String> fruits = new LinkedList<>();
```

```
    fruits.add("Apple");
```

```
    fruits.add("Orange");
```

```
    fruits.add("Mango");
```

```
    System.out.print("Queue: " + fruits);
```

```
    System.out.println("Queue: " + fruits);
```

```
    String display = fruits.peek();
```

```
    System.out.println("Is the Queue is empty: " + fruits.isEmpty());
```

```
    fruits.clear();
```

```
}
```

```
}
```

Deque :-

```
import java.lang.*;
```

```
import java.util.LinkedList;
```

```
import java.util.ArrayDeque;
```

Class Main {

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


ArrayDeque <String> args) {

fruits.add("Apple");

fruits.add("Banana");

fruits.add("Orange");

fruits.addFirst("orange");

fruits.addLast("Mango");

System.out.println("Deque: " + fruits);

String s = fruits.remove();

System.out.println("Deque: " + s);

String display = fruits.peek();

System.out.println("Deque: " + display);

boolean e = fruits.isEmpty();

fruits.clear();

System.out.println("Empty Deque: " + fruits);

boolean e1 = fruits.isEmpty();

System.out.println("Is the deque Empty: " + e1);

}

}

Hash map :-

```
import java.util.*; Map;  
import java.util.*; HashMap;
```

```
Class Main {
```

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

```
        Map<Integer String> fruits = new HashMap<>();
```

```
        fruits.put(1, "Apple");
```

```
        fruits.put(2, "Orange");
```

```
        fruits.put(3, "Mango");
```

```
        System.out.println("Map" + fruits);
```

```
        System.out.println("Keys: " + fruits.keySet()fruits.keySet()());
```

```
        System.out.println("Values" + fruits.values());
```

```
        System.out.println("Entries: " + fruits.entrySet());
```

```
        boolean value = fruits.remove(2, "Orange");
```

```
        System.out.println("Removed value: " + value);
```

```
        System.out.println("New Map: " + fruits);
```

```
    }
```

```
}
```