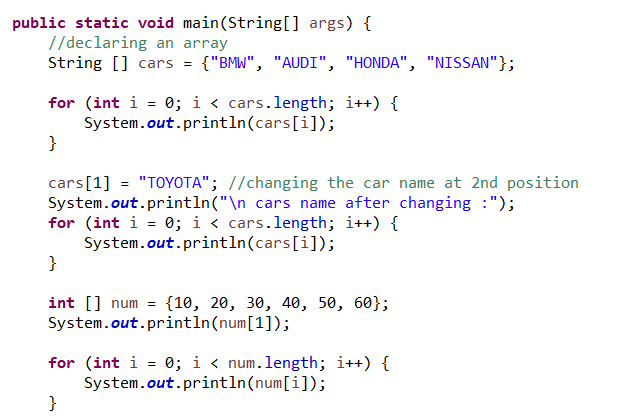
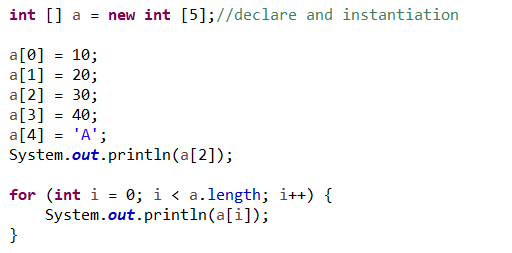
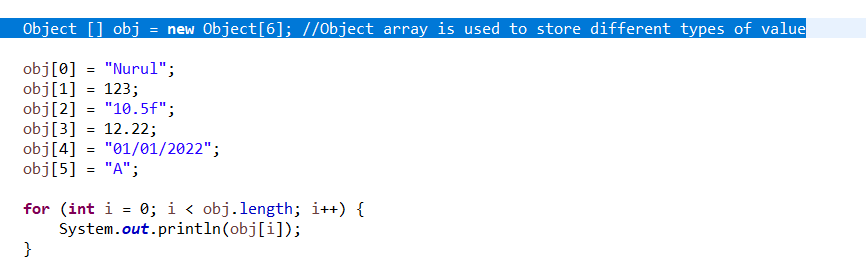
**🡸ARRAY🡺**

* **To store similar data type values in array variable;**
* **Arrays are used to store multiple values in a single variable, instead of declaring separate variables for each value.**
* **Lowest bound/index=0 and highest bound/index=(n-1) (n is size of array).**
* **Advantage: One variable can store multiple value. No, need to declare a lot of variables of same type data**
* **Disadvantage: Size is fixed---static array--to overcome this we use collections--Arraylist, Hash Table**
* **Store similar data types value---to overcome this we use object array.**
* **Random access: We can retrieve any data from array with the help of the index value.**
* **Add elements with the help of assignment operator.**
* **Does not support generic i.e., not type safe.**

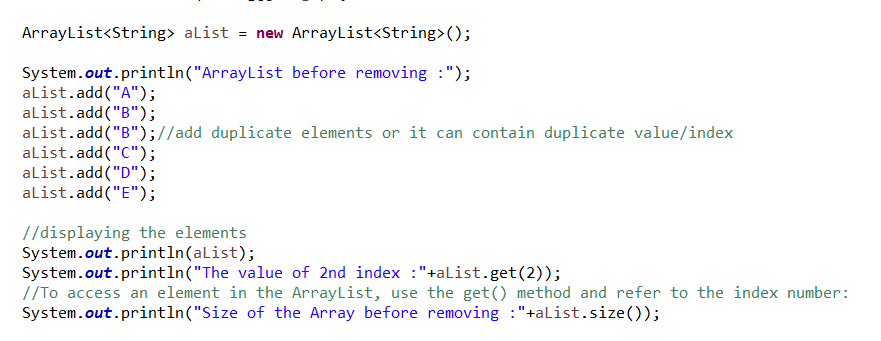
****

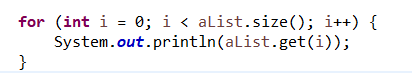
****

****

**🡸ARRAYLIST🡺**

* **Arraylist---dynamic array---we don't define the size.**
* **The ArrayList class implements the List interface.**
* **It uses a dynamic array to store the duplicate element of different data types.**
* **The ArrayList class maintains the insertion order and is non synchronized.**
* **Allows random access to fetch the element because it stores values on the basis of indexes. only store one object.**
* **Arraylist allows you to remove elements, but array doesn't provide such methods.**

****

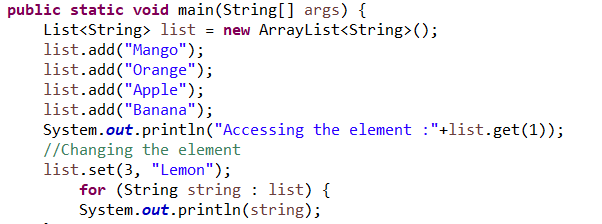
****

**Q. Difference between length of Array and size() of ArrayList?**

Array has length property which provides the length or capacity of the Array. It is the total space allocated during the initialization of the array. The size() method of ArrayList provides the number of objects available in the collection.

**🡸LIST🡺**

* **List is an interface cannot be instantiated. List interface extends collection frame work.**
* **The implementation classes of List interface are ArrayList, LinkedList, Stack, Vector.**
* **List maintain order of insertion and allows duplicate.**

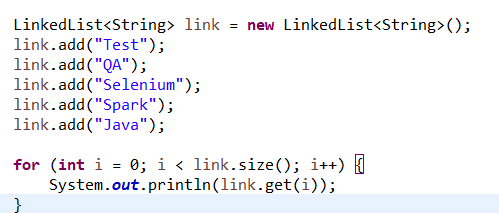
****

**🡸LINKEDLIST🡺**

* **Linked List is a class which is implementing list interface also deque interface.**
* **A linked list can also be defined as the collection of the nodes in which one node is connected to another node.**
* **Each and every node is divided into two parts. 1st parts store the value and 2nd part stores the reference of the next element.**
* **Each and every node has reference of next element that’s why it’s called singly linked list if there is a requirement of frequent addition and deletion in application then LinkedList is a best choice.**

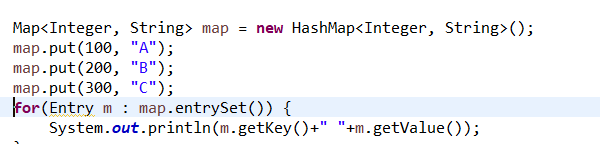
**Q. Types of Linked list:**

1. Singly Linked list
2. Doubly Linked list
3. Circular Linked list
4. Doubly Circular Linked list



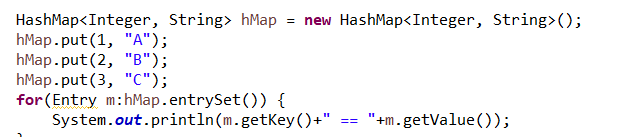
**🡸MAP🡺**

* **Map is an Interface and a map contains values on the basis of key, i.e. key and value pair. Each key and value pair is known as an entry. A Map contains unique keys.**
* **A Map is useful if you have to search, update or delete elements on the basis of a key.**
* **Java Map Hierarchy:**
* **There are two interfaces for implementing Map in java: Map and SortedMap, and three classes:**
* **HashMap, LinkedHashMap, and TreeMap.**
* **A Map doesn't allow duplicate keys, but you can have duplicate values.**
* **HashMap and LinkedHashMap allow null keys and values, but TreeMap doesn't allow any null key or value.**

****

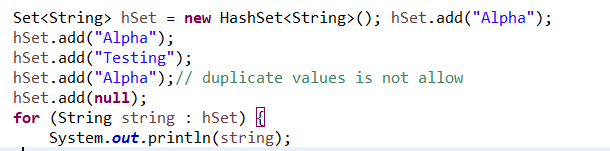
**🡸HASHMAP🡺**

* **HashMap is a class implements Map interface and extend AbstractMap.**
* **HashMap stores two objects key and value and HashMap doesn't allow duplicate key.**
* **It may have one null key and multiple null value and it maintains no order.**
* **HashMap is non-synchronized i.e., not thread safe.**
* **Concurrent Modification Exception---Fail Fast condition that means when one thread T1 is adding or removing the value and at the same time T2 T3 T4 try to access particular value but that value is gone from that object immediately it gives you CME.**

****

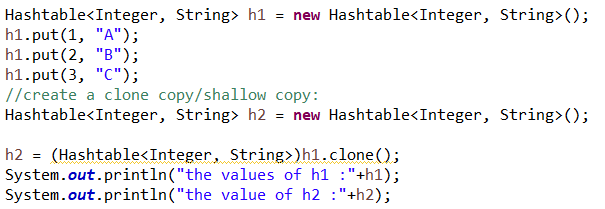
**🡸HASHSET🡺**

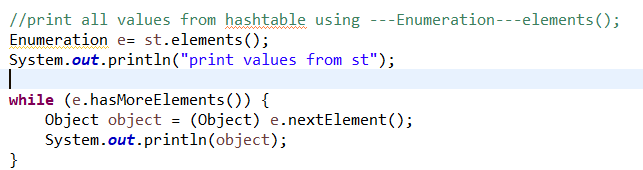
* **HashSet is a class that inherits the Abstract Set class and implements Set interface.**
* **HashSet stores the elements by using a mechanism called hashing.**
* **HashSet contains unique elements only and allows null value.**
* **HashSet class is non synchronized. HashSet is the best approach for search operations**
* **HashSet doesn't maintain the insertion order. Here, elements are inserted on the basis of their hash code.**

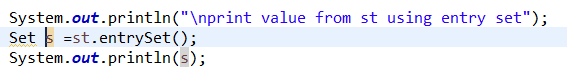


**🡸HASHTABLE🡺**

* **Java Hashtable class implements a hashtable. It inherits Dictionary class and implements the Map interface.**
* **Store the value on the basis of key-value and doesn't allow null key or value & contains unique elements.**
* **key-->object--hashcode--->value; when object is create, java provide a particular number for the object it is called Hashcode and it like 32 bit.**
* **The initial default capacity of Hashtable class is 11 whereas loadFactor is 0.75.**

****

****

****

**🡸Collection Vs Collections🡺**

**Collection:** Collection is a [**interface**](https://www.geeksforgeeks.org/interfaces-in-java/) present in **java.util.package**. It is used to represent a group of individual objects as a single unit. The**add()**,**remove()**, **clear()**, **size()**, and **contains()** are the important methods of the Collection interface.

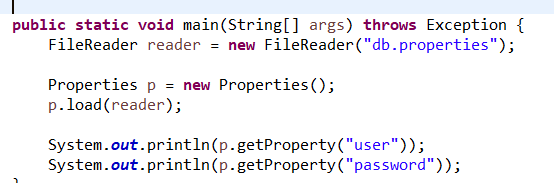
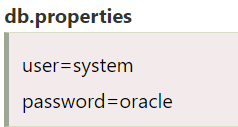
**Collections:** Collections is a utility **class** present in **java.util.package**. It defines several utility methods like sorting and searching which is used to operate on collection. It has all static methods.

🡸ITERATOR INTERFACE🡺

* **Iterator is an Interface to iterate over a collection.**
* **The user can apply these iterators to any of the classes of the Collection framework.**
* **In Java Iterator, we can use both of the read and remove operations.**
* **If a user is working with a for loop, they cannot modernize(add/remove) the Collection, whereas, if they use the Java Iterator, they can simply update the Collection.**
* **The Java Iterator is considered the Universal Cursor for the Collection API.**

**🡸PROPERTIES CLASS🡺**

**The properties object contains key and value pair both as a string. The java.util.Properties class is the subclass of Hashtable. It can be used to get property value based on the property key. The Properties class provides methods to get data from the properties file and store data into the properties file. Moreover, it can be used to get the properties of a system.**

**** ****



**🡸COMPARABLE AND COMPARATOR🡺**

* Comparable and Comparator both are interfaces and can be used to sort collection elements.

|  |  |
| --- | --- |
| **COMPARABLE** | **COMPARATOR** |
| Comparable provides a **single sorting sequence**. In other words, we can sort the collection on the basis of a single element such as id, name, and price. | The Comparator provides **multiple sorting sequences**. In other words, we can sort the collection on the basis of multiple elements such as id, name, and price etc. |
| 2) Comparable **affects the original class**, i.e., the actual class is modified. | Comparator **doesn't affect the original class**, i.e., the actual class is not modified. |
| 3) Comparable provides **compareTo() method** to sort elements. | Comparator provides **compare() method** to sort elements. |
| 4) Comparable is present in **java.lang** package. | A Comparator is present in the **java.util** package. |
| 5) We can sort the list elements of Comparable type by **Collections.sort(List)** method. | We can sort the list elements of Comparator type by **Collections.sort(List, Comparator)** method. |