***Array***

* An array is a container object that holds a fixed number of values of a single type.
* The length of an array is established when the array is created.
* After creation, its length is fixed.

Types of array

1.One dimensional array

2.Two dimensional array

3. Multi-dimensional array

* Creating an array
* dataType[] arrayRefVar = new dataType[arraySize];
* dataType[] arrayRefVar = {value0, value1, ...,valuek};

Example:

* double[] myList = new double[10];
* Array Resizing
* You cannot resize an array.
* You can use the same reference variable to refer to an entirely new array, such as:
* int[] myArray = new int[6];
* myArray = new int[10];

***Collections***

A collection is a single object managing a group of objects known as its elements.

Collection – A group of objects called elements;

Implementations determine whether there is specific ordering and whether duplicates are permitted.

* Set – An unordered collection; no duplicates are permitted.
* List – An ordered collection; duplicates are permitted.

***Map***

A Map object describes mappings from keys to values:

• Duplicate keys are not allowed

• One-to-many mappings from keys to values is not permitted

• The contents of the Map interface can be viewed and manipulated as collections

• entrySet – Returns a Set of all the key-value pairs.

• keySet – Returns a Set of all the keys in the map.

• values – Returns a Collection of all values in the map.

***Classic Collections***

The Vector class, which implements the List interface.

• The Stack class, which is a subclass of the Vector class and supports the push, pop, and peek methods.

• The Hashtable class, which implements the Map interface.

• The Properties class is an extension of Hashtable that only uses Strings for keys and values.

• Each of these collections has an elements method that returns an Enumeration object.