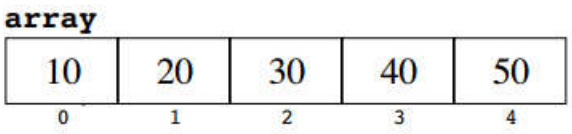
Standard Arrays
Create an arrays of integers of size 5
int[] myArray = new int[5];

Arrays are indexed starting at 0
Members of an array are accessed through the index
myArray[0] = 10
myArray[1] = 20
myArray[2] = 30



Example: Create an array called dblarray that takes 20 elements of type double
double[] dblarray = new double[20];

Create an array called slist that takes 15 elements of type String
String[] slist = new String[15];

The ArrayList
Standard arrays are not very flexible because they are not dynamic This means that in order to add an element to an array, one needs to create a new one, copy over the old elements and then destroy the old array. Often times we do not even know how big we want the array to be. For this reason, Java programmers prefer to use a java collection called the **ArrayList**

You can create an ArrayList of strings
ArrayList<String> names = new ArrayList<String>();

an ArrayList can also take integers
ArrayList<Integer> nams = new ArrayList<Integer>();

Or doubles
ArrayList<Double> names = new ArrayList<Double>();

or objects
ArrayList<Container> shapes = new ArrayList<Container>();

Assuming of course that the object 'Container' has been created
The data type must be specified during creation of the ArrayList
The example program blow a WHILE loop, whereby a user is prompted to enter names into an ArrayList. This is repeated until he enters a blank "", upon which the number of entries in the array, and its elements are listed and the program breaks. Note that in order to use an ArrayList collector, one must first import the **java.util.ArrayList** library

Example **Arrays.java**

```
import java.util.ArrayList;
import java.util.Scanner;

public class Arrays {

      public static void main(String[] args) {
            Scanner reader = new Scanner(System.in);
            ArrayList<String> names = new ArrayList<String>();
            System.out.println("enter blank to quit:");

             while (true) {
                  System.out.print("enter a name: ");
                  String name = reader.nextLine();
                  if (!name.equals("")) {
                        names.add(name);
                        System.out.println("\nadded "+name+" to names");

                  }else{
                        break;
                  }
             }
             System.out.println(names.size()+" entries created in array:");

             for(int i=0; i< names.size(); i++){
                   System.out.println(names.get(i));

             }

      }
}
```

Some methods in the ArrayList class are

add()	adds an element
size()	returns the size of an element
remove()	removes first instance of an element
void clear()	clears the array
get()	gets the element at the specified index
indexOf()	returns the index of a specified element
set()	sets the element at the specified index and returns the old value