1.

Behavior	Array	ArrayList
Create a reference to a list	int[] numbers = new int[6];	ArrayList<Integer> numbers = new ArrayList<Integer>(6);
Add an element to the end of the list [NOT REPLACE]	N/A – Arrays are fixed length and therefore there is no native way to change length	numbers.add(6);
Add an element to the middle of the list [NOT REPLACE]	N/A	numbers.add(3, 5);
Add an element to the beginning of the list [NOT REPLACE]	N/A	numbers.add(0,5);

Print a list	for(int val :numbers){
	System.out.print("Your values are");
	System.out.println(val);
	}
		for(int val :numbers){
		System.out.print("Your values are");
		System.out.println(val);
		}

Remove a specific item from a list	N/A	numbers.remove(3);
Compute the size of the list	int arraySize = numbers.length;	int arrayListSize = numbers.size();
Double the number of items the list can hold	N/A	int count; ArrayList<Integer> numbers = new ArrayList<Integer>(count); count * 2;

Convert an ArrayList to a regular Array	N/A	int array[] = new int[numbers.size()];
		for(int j=0;j<numbers.size();j++){
		array[j] = numbers.get(j);
		}

2. The increaseSize method is a method specifically made to address arrays shortcomings. Arrays are not natively able to change their size. Therefore, when the user needs to increase the size of their CD collection they need a way to increase the collection array. The increaseSize method does this by first creating a new array called temp. The temp array is given a size equal to twice that of the original collection array. Then a for loop is utilized to manually copy each value from the original collection array to the new temp array. An integer named cd is set to zero. The for loop is told to iterate as long as cd is less than the length of the original collection array. Each iteration will increment the cd variable so that the loop will terminate when it reaches the length of the original array. Both the old collection array and the new temp array are told to use the value of cd for their index value each time the for loop iterates. The new temp array is set equal to the collection array. This essentially allows the for loop to step through each of collection's values, copying each one to the new temp array at each of their respective indices. After the for loop completes it then assigns collection as a reference to the new temp array which contains all of the old values plus room for twice the values it held before.

3. Many would say that ArrayLists are generally better and more versatile than Arrays, but each has its purpose. There are many differences between the two

I. Resizable – Array are static in size, and their length cannot be changed after creation. However Arraylists are dynamic in size, growing to the capacity needs of their users automatically. Arraylists have the ArrayList.add(), ArrayList.remove(), and ArrayList.set() methods which all are incredibly useful and lacking from arrays.

II. Dimensions – Arrays can have multiple dimensions, while ArrayLists are always single dimensional. For example

int array[][] = new int[5][6]

ArrayList<Integer> numbers = new ArrayList<Integer>(6);

Multiple dimensions can be very useful in programming.

III. Performance – performance of Array and Arraylist is dependent on the code it is running. Automatic resizing of ArrayList slows down performance as it uses a temporary array to copy elements from the old array to a new array.

All in all ArrayLists are the much better choice if you are making a list that in any way needs to have its size changed, or insert values at specific indices. Otherwise it would most likely be better to just use an Array.

4.

A.This line would remove the String “Pete” from the band Arraylist. Each string below this one would be moved up and have its index changed respectively.

B. A regular array of strings would not be able to have a value removed this easily. You would need to create another array, copy only the values you want to keep, then set your old array variable to reference the new updated array you’ve created.

C. HashMaps allow you to assign one index multiple values. Arraylists can only store one value per index.

D. A List is an ordered Collection (sometimes called a sequence). Lists may contain duplicate elements.

E. The Java platform includes a collections framework. A collection is an object that represents a group of objects (such as the classic Vector class). A collections framework is a unified architecture for representing and manipulating collections, enabling collections to be manipulated independently of implementation details. It includes the ArrayList and HashMap classes.

Simple demonstration of Collections framework

```
/*
Java HashMap example
*/

import java.util.HashMap;

public class JavaHashMapExample {

  public static void main(String[] args) {

    HashMap hMap = new HashMap();

    hMap.put("One", new Integer(1));
    hMap.put("Two", new Integer(2));
```

```
    Object obj = hMap.get("One");
    System.out.println(obj);


  }
}
/*
Output of the program would be
1
*/
```