ArrayList:

In Java, an `ArrayList` is a dynamic array that allows you to store and manipulate a collection of elements. Unlike regular arrays, `ArrayList` can grow or shrink in size dynamically as you add or remove elements. It is part of the Java Collections Framework and is defined in the `java.util` package.

Here's how you can use `ArrayList` with two code examples:

Example 1: Creating and Using an ArrayList

In this example, we will create an `ArrayList`, add elements to it, retrieve elements, and perform some common operations:

```java
import java.util.ArrayList;

public class ArrayListExample {
    public static void main(String[] args) {
        // Create an ArrayList of Strings
        ArrayList<String> fruits = new ArrayList<>();

        // Add elements to the ArrayList
        fruits.add("Apple");
        fruits.add("Banana");
        fruits.add("Orange");
        fruits.add("Grapes");

        // Get the size of the ArrayList
        int size = fruits.size();
        System.out.println("Size of the ArrayList: " + size);

        // Access elements by index
        System.out.println("First fruit: " + fruits.get(0));
        System.out.println("Second fruit: " + fruits.get(1));

        // Check if an element exists in the ArrayList
        boolean containsMango = fruits.contains("Mango");
        System.out.println("Contains Mango? " + containsMango);

        // Remove an element by value
        fruits.remove("Banana");
        System.out.println("After removing Banana: " + fruits);
        // Iterate through the ArrayList
        System.out.println("Fruits in the ArrayList:");
        for (String fruit : fruits) {
            System.out.println(fruit);
        }
    }
}
```

Example 2: ArrayList with Different Data Types

In this example, we will use an `ArrayList` to store elements of different data types:

```java
import java.util.ArrayList;

public class MixedArrayList {
    public static void main(String[] args) {
        // Create an ArrayList to hold mixed data types
        ArrayList<Object> mixedList = new ArrayList<>();

        // Add elements of different data types
        mixedList.add("Hello");
        mixedList.add(42);
        mixedList.add(3.14159);
        mixedList.add(true);

        // Access elements from the mixed ArrayList
        for (Object item : mixedList) {
            System.out.println(item);
        }
    }
}
```

In this example, we use an `ArrayList<Object>` to store elements of different data types. While it's possible, it's important to be cautious when working with mixed data types in an `ArrayList`, as you may need to perform type-checking or type-casting when retrieving elements.

Remember to import the `ArrayList` class from the `java.util` package when using it in your Java code.