ArrayList

The ArrayLists are similar to Arrays in many ways, but they have the ability to change size during the execution of your program. They also have many useful helper methods for operating on them and the data they contain.

Importing ArrayList

ArrayList needs to be imported in order to use it!

```
//import all classes in the java.util package
import java.util.*;
//import only ArrayList
import java.util.ArrayList;
```

Creating an ArrayList

3 Constructors you need to know

```
//Create an empty ArrayList
ArrayList empty = new ArrayList();
//Create an empty ArrayList with set size
ArrayList sized = new ArrayList(10);
//Create an array list from another array list
ArrayList cloned = new ArrayList(sized);
```

Generics

Generics allow you to specify the type of the data your list will contain.

```
//Generic type specified for variable and constructor
ArrayList<String> stringList = new ArrayList<String>();
//Generic type only specified for variable
ArrayList<String> stringListShort = new ArrayList<>();
```

Add

the `add()` method allows you to insert a new value

```
//insert at end
list.add("New End");
//insert at index
list.add(5, "Insert at index 5");

//java will not compile if you add a bad type!
ArrayList<String> stringList = new ArrayList<>();
//Boolean.TRUE is not a String!
//Will not commpile!
stringList.add(Boolean.TRUE)
```

Remove

the `remove()` method allows you to remove values

```
//remove by matching object
list.remove("New End");
//remove by matching index
list.remove(0)
```

Set

the `set()` mehtod allows you to update an element in the list

```
//update by index
list.set(0, "Replace index 0 with this value");
//you cannot update elements that do not exist
//this will result in a runtime exception, not a compile e
list.set(1000, "Can't do this!");
```

Size and isEmpty

the `size()` method returns the size of the list

the `isEmpty()` method returns true if list is empty

Clear

the `clear()` method discards all elements from the list

```
list.add("item one");
list.add("item two");
//outputs 2
list.size();
//clear the list
list.clear();
//outputs 0
list.size();
```

Contains

the `contains()` method checks whether a value is in a list

```
list.add("item one");
list.add("item two");
//outputs true
list.contains("item one");
//outputs false
list.contains("item three");
```

Equals

the `equals()` method checks whether 2 lists have the same elements in the same order

```
ArrayList<String> list1 = new ArrayList<>();
ArrayList<String> list2 = new ArrayList<>();
ArrayList<String> list3 = new ArrayList<>();

list1.add("pizza");
list1.add("burger")

list2.add("pizza");
list2.add("burger")

list3.add("burger");
list3.add("pizza")

//outputs true
list1.equals(list2);

//outputs false
list1.equals(list3);
```

Homework:

Create a program to manage a movie collection

Step One

Implement a Movie class with a String Title

The constructor should work like this:

```
//Create a new movie with title "The Last Jedi" and ratin
Movie starWars = new Movie("The Last Jedi", 5)
```

Step Two

Create a list of movies using the ArrayList constructor

Step Three

Demonstrate Adding movies with the `add()` method
Demonstrate Removing movies with the `remove()` method
Demonstrate Updating a movie with the `set()` method

Step Four

Using methods in combination

Demonstrate using `contains()` and `add()` together to only add a movie if it is not already present
Demonstrate using `set()` and `isEmpty()` to only update a move if the list isn't empty
Demonstrate using `set()` and `size()` to only update a movie by index if that index is present

Additional Resources

- Pluralsight Java Fundamentals Core - Chapter 6 Collections
- Pluralsight Java Fundamentals - Chapter 5 (review arrays)
- ArrayList Docs
- MIT ArrayList Slides