IEEE
Computer Chapter
Southern Alberta Section

SCHULICH
School of Engineering
UNIVERSITY OF CALGARY

IEEE Women in Engineering
WIE
Southern Alberta Section

SCHULICH IGNITE 2019

Extra Slides

- What is an ArrayList
- Making an ArrayList
- Using an ArrayList

What is an ArrayList?

What is an ArrayList

An ArrayList are a different data type
It is like a more complicated versions of arrays

Unlike arrays, ArrayLists **can change size**. You can add and remove stuff easily.

- This works great if you have a lot of variables, but you don’t know exactly how many

Making an ArrayList?

How to Make an ArrayList

To make an ArrayList for String objects, type in:

```
ArrayList<String> myList = new ArrayList<String>();
```

To make an ArrayList for MyBall objects, type in:

```
ArrayList<MyBall> ballList = new ArrayList<MyBall>();
```

To make an ArrayList for **integers** or **floats**, you need something else...

ArrayLists and Primitive Data Types

This will **NOT** work:

```
ArrayList<int> myList = new ArrayList<int>();
```

That's because int, float, boolean, char are **primitive data types**. All of them are **lowercase**

The opposite of primitive data types are **objects**. Examples of these data types (same thing as classes) are:

- String, Ball, Player, Bullet

All classes should start with an **Uppercase**

ArrayLists and Primitive Data Types

To get ArrayLists of primitive data types, use:

```
ArrayList<Integer> intList = new ArrayList<Integer>();

ArrayList<Float> floatList = new ArrayList<Float>();

ArrayList<Boolean> boolList = new ArrayList<Boolean>();
```

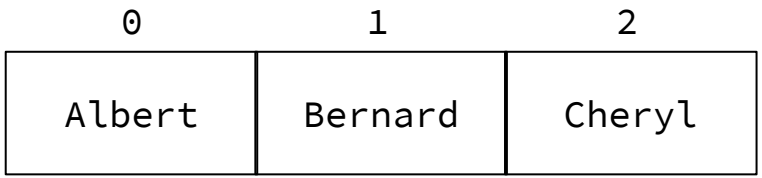
The classes `Integer`, `Float`, and `Boolean` act just like the primitives `int`, `float`, and `boolean` in almost all other ways, so use them like normal.

Using an ArrayList

Adding to an ArrayList

To add to the ArrayList, use list.**add()** function.

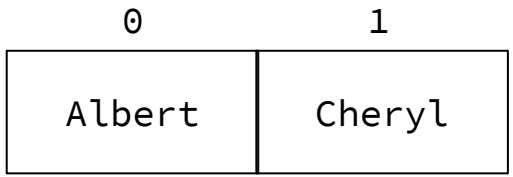
```
ArrayList<String> myList = new ArrayList<String>();
myList.add("Albert");
myList.add("Bernard");

String name = "Cheryl";
myList.add(name); // You can add variables too!
```



Removing From an ArrayList

To add to the ArrayList, use list.**remove()** function.

```
myList.remove("Bernard");
```



Getting One Value From the ArrayList

To get a value from the ArrayList, use list.**get()**

```
String name = myList.get(0);
println(name);   // Prints out "Albert"
```

Use list.**size()** instead of length

```
for (int i = 0; i < myList.size(); i++) {
    println(myList.get(i));
}
```

Looping Over Arrays

You can use a special syntax called **for-each** loops to quickly loop over every element in the ArrayList

```
for (String name : myList) {
    println(name);
}

// Prints out "Albert", "Cheryl"
```

This also works with regular arrays