IEEE
Computer Chapter
Southern Alberta Section

SCHULICH
School of Engineering
UNIVERSITY OF CALGARY

IEEE Women in Engineering
WIE
Southern Alberta Section

SCHULICH IGNITE 2019

Session Overview

- Intro to **classes** and **objects**
- Intro to **arrays**

Exercise 1: Lotsa Balls

Make 5 balls, each with different sizes, positions, and speeds, and directions.

Let’s both do it in **30 seconds**,

GO!

Classes and Objects

What's an object?

- It's a user defined variable!
- Can group several variables together
- Here is an object called myBall:

```
Ball myBall;
```

So how do we create a new type?

- With **classes**!
- **Classes** are the way we describe our new type of variable to the computer
- (They do more than that, ask your mentors!)

Example

```
class Ball {
    int x;
    int y;
    float speedX;
    float speedY;
}
```

- Note: Type **Ball**, not ball (capitalization matters!)

How do we use them?

```
Ball myBall = new Ball();
myBall.x = 100;
myBall.y = 200;
myBall.speedX = 10;
myBall.speedY = 5
```

Exercise 2: Lotsa Balls (Better)

Make 5 balls, each with different sizes, positions, and speeds, and directions.

BUT now with objects!

(Hint: start by making **class Ball**)

Arrays

What's an array?

- An **array** is a list of data - handy for storing a lot of whatever you want!
- It allows you to create a whole bunch of similar variable at the same time
- Lets us reduce the number of separate variables we have to declare

How do I make room?

- [] is the tells the computer this is an **array**
- Example below creates a new array of ints called myNumbers

```
int[] myNumbers = new int[4];
```

- The array myNumbers can carry 4 ints

Arrays continued

Arrays data is accessed via an **array index**

Index	**0**	**1**	**2**	**3**
Value	101	202	303	404

To get the **third** value in the array, you would need to use index **2**

```
println( myNumbers[2] );
```

Getting array elements

```
int[] myNumbers = new int[4];

myNumbers[0] = 101;
myNumbers[1] = 202;
myNumbers[2] = 303;
myNumbers[3] = 404;
```

Bah, that's too much work!

- A way of filling values of the array without having to do a lot of work

```
int[] myNumbers = { 101, 202, 303, 404 };
```

- IMPORTANT: This only works when creating an array

```
myNumbers = { 55, 66, 77, 88 };  // Not a chance
```

Arrays and Sizes

- Arrays have a predefined variable called **length** - This is the size of the array
- What will print out to the console?

```
int[] bigArray = new int[400];


println(bigArray.length);
```

- Note: Array sizes can't be changed after they're made!

Exercise 3: Different Arrays

Set aside the exercise from before,
we'll be using it again

- Make an array of Strings
  (don't forget the capital S)
- Print them all out to the console
- Print them all out using a variable as the index
- Make it so that it works with any array size
  (hint: use .length)

Exercise 4: Lotsa Balls (the best way)

Finish the exercise from before, but this time using an array of Balls to make your life a little easier

- Make 5 balls, each with different size, position, speed and direction
- Hint: you need to write **new Ball()** 5 times