Recommended Development Tools

- Android Studio
  - https://developer.android.com/sdk/installing/studio.html
- Genymotion (Android Emulator)
  - http://www.genymotion.com/

Android 101

Frederick Widjaja

Java

Java

- Very similar in syntax to C
- Object-oriented…

- Types

C	Java
short, int, long	byte (8), short (16), int (32), long (64)
char	char (16)
bool (int)	boolean
float, double	float (32), double (64)

Java

```
for (int i = 0; i < 10; i++) {
        System.out.println("Count: " + i);
}


int j = 0;
while (j < 3) {
        System.out.println("Count: " + j);
        j++;
}


if (x > 2 && y > 3) {
        …
} else if (z != 0 || w == 1) {
        …
} else {
        …
}
```

Objects

- Real-world objects share two characteristics
  - State
  - Behavior

  - Dogs
    - States: Name, Color, Breed, Hungry
    - Behavior: Bark, Wag tail, Fetch ball
  - Cars
    - States: Model, Year, Max speed, Current Speed
    - Behavior: Accelerate, Brake, Reverse
- Java Objects have
  - Fields (State)
  - Methods (Behavior)

Classes

- Java Classes are the “blueprints” for an Object

```
public class Car {
        private String model;
        private int year;
        private double speed;

        public Car(String model, int year) {
                this.model = model;
                this.year = year;
        }

        public void accelerate(double increment) {
                speed += increment;
        }

        public void brake() {
                speed = 0;
        }

        public double getSpeed() {
                return speed;
        }
}
```

Classes

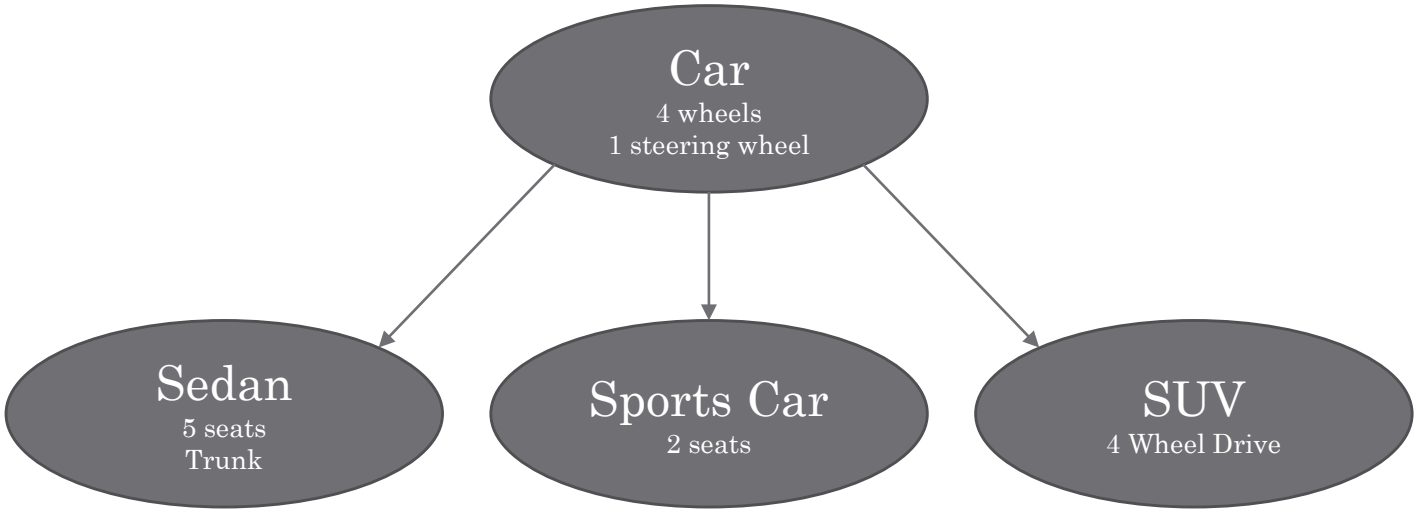
```
public class Main {
	public static void main(String[] args) {
		Car car = new Car("Prius", 2014);
		car.accelerate(60.0);

		System.out.println("Speed: " + car.getSpeed());

		car.brake();
		System.out.println("Speed: " + car.getSpeed());
	}
}
```

Inheritance

- Different Objects may share common characteristics with each other.
- Object-Oriented Programming (OOP) allows classes to inherit commonly used states and behaviors from other classes.



Inheritance

```
public class Sedan extends Car {
        private double maxSpeed;
        private int maxCapacity;
        private int numPassengers;

        public Sedan(String model, int year, double maxSpeed) {
                super(model, year);
                this.maxSpeed = maxSpeed;
                this.maxCapacity = 5;
                this.numPassengers = 0;
        }

        @Override
        public void accelerate(double increment) {
                if (getSpeed() + increment <= maxSpeed) {
                        super.accelerate(increment);
                }
        }

        public void setNumPassengers(int numPassengers) {
                this.numPassengers = numPassengers;
        }

        public int getNumPassengers() {
                return numPassengers;
        }
}
```

Interfaces

- Exposes methods to the outside world
  - Abstracts away complexities
  - “Black Box”

```
interface Bicycle {
        void speedUp(int increment);
        void brake();
        void changeGear(int newGear);
}

class MountainBike implements Bicycle {
        int speed = 0;
        int gear = 1;

        void speedUp(int increment) {
                speed += increment;
        }

        void brake() {
                speed = 0;
        }

        void changeGear(int newGear) {
                gear = newGear;
        }

        …
}
```

More on Java

- The Java Tutorials
  - https://docs.oracle.com/javase/tutorial/
- Java API Documentation
  - https://docs.oracle.com/javase/7/docs/api/

Android

Jargon

- Activity
  - A single screen within an application. Usually limited to one single “activity” (e.g. Viewing, Adding, Editing)
- View
  - An object that draws to a rectangular area on the screen and handles clicks, keystrokes, and other interaction events.
- Intent
  - A “message” you can use to launch or communicate with other applications/activities.
- Manifest File
  - An XML file that each application must define that gives information about the application itself.
    - Version
    - Activities

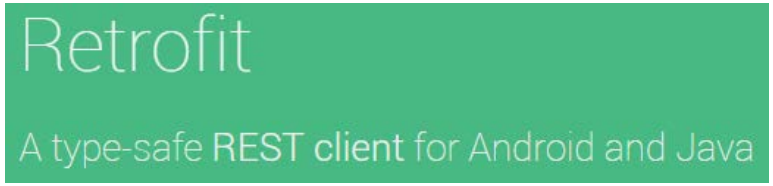
Let’s Code!

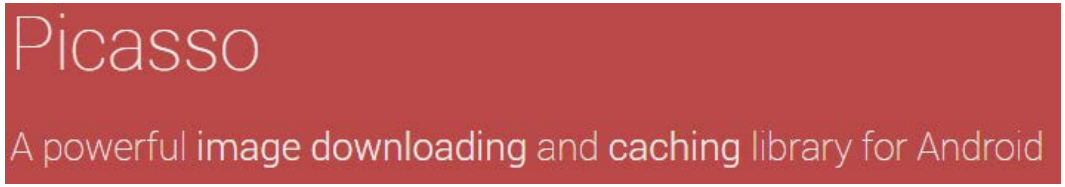
More on Android

- Android Tutorials
  - https://developer.android.com/training/index.html
- Android API Guide
  - https://developer.android.com/guide/index.html
- Android Documentation
  - https://developer.android.com/reference/packages.html
- Another Android Tutorial
  - http://www.vogella.com/tutorials/Android/article.html

Libraries

- Extend the functionality of your apps with less code!





Happy Coding!