Android 101:

Build an app from start to finish

Intro to Android and Java

What is Java?

Object Oriented Language

Platform Independent

Compiled into "executable"

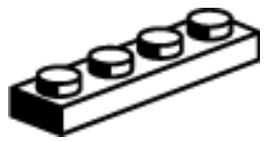
Object Oriented Programming

Everything you build is an object

Each object can have data

Each object can have behavior

Objects



Everything you build is an object

Data -- "state"

"State" is data about an object

Legos:

- Color
- Size
- Connectors
- Collection

Giraffes:

- Color
- Height
- Spots
- Hunger Level

Behavior -- "methods"

A "Method" is a set of instruction to perform an action

Legos:

- Stack on top
- Stack below
- Stack sideways

Giraffes:

- Walk
- Reach for leaf
- Eat leaf

Structure

Method:

piece of code that are instructions for behavior

Class:

file that has multiple methods

All the giraffe instruction would like be in a class called Giraffe

Library:

lots of classes

The SDK and JDK you downloaded are libraries

Public and Private

Private and Public:

methods can be public or private

Private: only other methods in the same class can call them

Public: any method in any class can call them

Returns

Methods can return data

A method that asks how tall a giraffe is would return

a number

A method that asks the name of a giraffe would return

a word

voids

A method that tells the giraffe would return nothing

it would just make the giraffe walk

this is called a VOID method

Variables

Hold any kind of data

numbers, words, giraffes, legos

Local variables

can be used in one method or one class

Global variables

can be used throughout application

Using variables

"Declaring" -- Saying that something is a variable

```
int x;

String y;

float z;
```

"Assigning" -- Giving variables a value

```
x=26; OR int x=26;

y="Hi, there"; OR String y = "Hi, there";

z= 6.01; OR float z= 6.01;
```

Variables and Math

+	Addition
–	Subtraction
*	Multiplication
/	Division
%	Modulus
++	Increment
––	Decrement

Variables and Math

```
int bananas;
int oranges;

bananas = 6;
oranges = 3;

int fruit = bananas + oranges;
```

Conditional Statements

```
if (fruit < 2){
  goBuyFruit();
}
else{
  eatFruit();
}
```

==	Equals?
>	Greater than?
<	Less than?
!=	Not Equal to?
&&	AND
||	OR

Loops

```
int i = 0;
while (i < 10){
  System.out.println(i);
  i++;
}


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

Android app structure

res

drawables

layout

values

src

com.companyname.appname

AndroidManifest.xml

AndroidManifest.xml

permissions

sdk version

activities

configurations

Layouts

LinearLayout vs. RelativeLayout

TextView

EditText

Button

Layouts

```
<LinearLayout
 xmlns:android="http://
 schemas.android.com/apk/res/android"
 android:layout_width="fill_parent"
 android:layout_height="fill_parent"
 android:orientation="vertical"
 android:padding="8dip"
 android:background="@color/WHITE">
```

Layouts

```
<TextView
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:text="Welcome text"
  android:textColor="@color/PURPLE"
  android:id="@+id/welcomeText"
  android:padding="16dip"/>
```

Layouts

```
<EditText
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:inputType="textCapSentences"
android:hint="Enter username"
android:id="@+id/usernameField"
android:textColor="@color/RED"/>
```

Layouts

```
<Button
 android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="Sign in"
android:onClick="login"
android:id="@+id/login"/>
```

Welcome Activity

Package lets us put all the code together

```
package com.girldevelopit.android.views;
```

```
import android.app.Activity;
import android.app.AlertDialog;
import ...
```

Import lets us use code from other classes or libraries

Welcome Activity

```
public class WelcomeActivity extends
Activity
{
    private GirlDevelopIt app;

    private EditText usernameField;
    private Button login;
    private Button logout;
    private TextView welcomeText;
```

Welcome Activity

```
@Override
public void onCreate(Bundle
savedInstanceState)
{
  super.onCreate(savedInstanceState);
  setContentView(R.layout.welcome);

  this.app =
  (GirlDevelopIt)getApplicationContext()
  ;

  initElements();
}
```

Welcome Activity

```
private void initElements(){
  usernameField =
  (EditText)this.findViewById(R.id.usernameField);

  login = (Button) this.findViewById(R.id.login);

  logout = (Button) this.findViewById(R.id.logout);

 welcomeText =
 (TextView)this.findViewById(R.id.welcomeText);
```

Welcome Activity

```
if (app.getUsername() == null ||
app.getUsername().equals("")){
    usernameField.setVisibility(View.VISIBLE);
    login.setVisibility(View.VISIBLE);

    welcomeText.setText("Please login to add images
    to the gallery");

    logout.setVisibility(View.GONE);
}
```

Welcome Activity

```
else{
  usernameField.setVisibility(View.GONE);

  login.setVisibility(View.GONE);

  welcomeText.setText("Welcome back, "+
  app.getUsername() + "!");

  logout.setVisibility(View.VISIBLE);
}
```

Welcome Activity

```
public void login(View view){
  if(usernameField.getText().toString().equals("")){

    AlertDialog.Builder builder = new AlertDialog.Builder(this);

    builder.setMessage("Please enter your username")
           .setCancelable(false)
           .setNegativeButton("OK", new
           DialogInterface.OnClickListener() {
             public void onClick(DialogInterface dialog, int id) {
                dialog.cancel();
              }
          });

  AlertDialog alert = builder.create();
   alert.show();
}
```

Welcome Activity

```
else{

  app.setUsername(usernameField.getText().toString())
  ;

  usernameField.setText("");

  initElements();

}
```

Welcome Activity

```
public void logout(View view){

  app.setUsername("");

  initElements();
}
```

Next Week!

Learning to connect Activities

Accessing the camera

Creating a class with Methods and States

Questions?