Girl Develop It, Class 2

- Android
  - Mobile history
    - Proprietary APIs, different for every phone out there
    - You had to write it 20 times to get it on 20 different types of phones!
    - Doesn't scale very well, lots of effort duplication.
    - Not easy, in a low-level programming language.
  - Why is Android special?
    - Open source (people can download, view, inspect, and edit the source code) & built on a Linux kernel
    - Many device types: phones, tablets, google TV
  - Why develop for Android?
    - No start up costs
    - No restrictions on distribution
    - No approval process
    - Contrast this with the iOS development world: You need to pay Apple $100 per year to get the app on a real phone. You have to go through a lengthy approval process and Apple tightly controls what you can do with the phone.
- Anatomy Lab
  - Android is a mixture of XML and Java. XML is where you specify what the view will look like and you put your logic into the Java files.
  - AndroidManifest.xml - Where you define the parts of your application.
    - Activities represent each screen in your app. You have to specify them here, otherwise the application will crash immediately (with a pretty informative error message compared to other ones I've seen).
    - Services are used for background processing.
    - Icon - The drawable icon that will represent your application
    - Theme - How your application will look to users
    - Version code is an integer that must increase on each release. This is how the market knows there is a newer version that has been published so that it will notify users to upgrade.
    - Version name is the public facing name that will be displayed in the market.
    - Requirements
      - Min, max, and target SDKs restrict which phones/tablets can download the app.
      - Required hardware (example phone, GPS, keyboard) can be specified and will be used to determine who can see your app in the market. This is so that if you know an app doesn't work well without a particular feature, you can limit it's distribution to those devices.
        If your application needs to make telephone calls, a wifi only device shouldn't be able to use your app.
      - Permissions provide access to device features and data. A user needs to explicitly agree to these permissions when downloading the app from the Google Play market. Make sure to balance features and access to private information to avoid creeping out your users.
        You can't have an internet browser with out internet access and a mapping programming would be useless without GPS.
  - src/ - Where you store all your source/Java code.
    - The package name you specified in the wizard matches file structure.
      com.colabug.calc -> src/com/colabug/calculator
    - You add source files here for activities, fragments, and utilities.
    - You can create sub directories to organize your files (because apps get huge).
  - res/ - Where you store resources like layouts, images, colors, and strings.
    - These resources are complied into the R.java file.

- Using Resources
- layout/
- values/
  - styles.xml
  - strings.xml
- drawable/ - Where you put images

- Widgets & Layouts
  - There are many different types of widgets at your disposal.
  - Widgets comprise a user interface. There many predefined widgets, including text labels, radio buttons, images, and web views. The best place to discover them is on developer.android.com.
  - Layouts are used to organize your widgets and lay them to as you want them to appear in the user interface.
  - You can nest layouts to create complex layouts.

    Example: A button and a text label are in a vertical linear layout, and then you place that in a horizontal layout to alight it with a checkbox on the right side of the screen.
  - You give widgets an id so that you can reference it in the Java code.

```
<!--In the layout/XML file-->
<TextView
    android:id="@+id/welcome_label"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/welcome_text"/>

// Java code
findViewById(R.id.welcome_label);
```

- Class project: Calculator