Training February 23th, 2016

0211 pass code for the training room

Install **(HW)**

* Android Studio
* Genymotion
  + Plug-in
* USB Driver (if needed) for ASUS Nexus Tablet
* GitHub

Presentation of everyone, who we are and why we choose Android.

Review **(HW)**

* Android JIT, ADT, Dalvik, ART, java runs in a JVM.

Homework **(HW)**

* Review all
* Prepare all
* Additional homework

Resume

* Bring your resume **(HW)**

Review **(HW)**

* OOP: Another level of abstraction creating objects that can be used without knowing the composition of it.
  + Encapsulation: Encloses all properties of an object.
  + Inheritance: Inherit all properties and methods from the superclass or parent.
  + Polymorphism: It can have many forms.
* Also
  + Interfaces: A contract for objects, by giving them an interface and knowing what is, you can make it do an action. Makes things easier to treat objects.
  + Abstract classes: A class that cannot be instantiated.
* Java
  + Collection framework & arrays
  + Generics
  + Threads **(HW)**
  + Lambda groups
  + Everything else
* Design patterns
  + MVC: Model View Controller
  + MVP: Model View Presenter
  + MVVM: Model View View Model
  + Singleton: You can just have one instance of a class.
  + Builder: Uses an object to create an instance of another object.
  + Observer: Breaks things into 2, you observe an object to see if they have information for you, and things are subscribed to you, if you have something to tell them you tell them.

What’s a good thing about design patterns and what’s a bad?

Mobile Apps

* Native Apps

&

* HTML5: It will runs the same everywhere.
* CSS3
* JavaScript

&

* PhoneGap
* Cordova: These two know how to communicate the mobile website to the phone.

Universal apps

* Xamarin
  + .NET
  + C#

Version control

* Step 1
* Step 2
* Step 3
* Git: created by the same guy who created Linux.
  + GitHub **(HW)**
    - Main Repo
    - Local Repo
    - Fork
    - Clone
    - Merge
    - Pull Request

Android Development

* Architecture of Android Apps
* IDE
* First Android app
  + Templates
  + Layout XML
  + Activity class
  + Resources
  + Genymotion
  + Running the App
  + Inflating the Layout
  + The Context
  + Referencing a view
  + Events
* Android resources
  + Strings
  + Colors
  + Measurements
  + Images
* Android Assets **(HW)**
* Android units of Measure

dp

* + sp
  + dpi
* Lint

Explicit intent

Implicit intent

Generics <>

Checking or errors before compilation. The errors marked by the IDEs.

AsyncTask send three types,