Compsci 290.3, Spring 2017
Software Design and Implementation: Mobile
Landon Cox
Owen Astrachan

http://www.cs.duke.edu/courses/spring17/compsci290.3

See also Sakai @ Duke for all information

Hello!

- **About me**
  - ➤ **Duke undergrad**
  - ➤ **Michigan PhD**
  - ➤ **Sabbatical at Facebook**
- **Research interests**
  - ➤ **OS, distributed systems**
  - ➤ **Privacy and security**
  - ➤ **Moving into robotics**

Hello!

- **About me**
  - ➤ **Schools with the letter 'D'**
  - ➤ **Lots of Duke Connections**
  - ➤ **Google v Oracle**
- **Research interests**
  - ➤ **Astrachan's Law**
  - ➤ **Teaching Computer Science K-12, College**
  - ➤ **Design Patterns resurfacing**

Where does 290.3 Fit?

- **Alternative to 308: Software Design and Implementation (Mobile)**
  - ➢ **Many concepts are the same**
  - ➢ **Platform is different**
  - ➢ **Many issues in mobile aren't on the desktop (more later)**
- **Fit into curriculum**
  - ➢ **Design and deliver in teams is good for …**
  - ➢ **Good design comes from experience, experience comes from bad design (attributed to many)**

Goals and Learning Objectives

- **Turn ideas and specifications into working, mobile Android applications**

- **Work collaboratively on a small team**

- **Leverage and demonstrate understanding of tradeoffs in creating code for mobile device(s) from software perspective**

- **Demonstrate understanding of privacy and security concerns related to mobile apps**

- **More …**

Administrivia

- **We will have readings for the course, these will be online**
  - ➢ **synched with 308 when feasible/possible**
- **Books completely optional, lots of material on the web, but books can be very useful**
- **Projects, groups, midterm, final**

Java Classes, Review

- **Class encapsulates state and behavior**
  - ➤ **State is typically private**
  - ➤ **Android uses `mInstanceVar` convention**
- **Class is an object factory**
  - ➤ **Calling new creates a new instance**
  - ➤ **Everything in Java is a pointer/reference**

Classes and objects

- **Classes communicate and collaborate**
  - ➤ **Parameters: use-a, send and receive**
  - ➤ **Containment: has-a, aggregate or responsible**
  - ➤ **Inheritance: is-a, extends, specializes**

- **Inheritance and interfaces**
  - ➤ **More on this throughout semester, especially at beginning**

Android Walkthrough

● **See android demo**