**Android Basics Summary**

http://developer.android.com/training/index.html

**Building Apps with Connectivity & the Cloud**

* **Connecting Devices Wirelessly** 
  + Can communicate with other devices on the same local network (just have to be physically nearby)
  + Network Service Discovery allows an application to seek out a nearby device running services with which it can communicate
  + Merchants can broadcast its app
  + Wi-Fi Direct
* **Performing Network Operations**
* Write applications that have fine-grained control over usage of network resources:
  + - Control app’s data habits
    - How often your app syncs data
    - Whether to perform uploads/downloads only when on Wi-Fi
    - Whether to use data while roaming
    - Etc…
* Only fetch large data when Wi-Fi network is available
  + - Reasons of faster than mobile data, and mobile data is metered
* Check for network connection FIRST (availability and/or connectivity)
* Parsing XML Data – for parsing Newsfeeds in apps, etc.
* **Transferring Data Without Draining the Battery**
* Optimizing downloads for efficient network access
* Bundled transfers lets you transfer data with less battery drain
* Prefetching data reduces number of independent data transfer sessions
  + - Downloads all data you are likely to need for a given time period in a single burst, over a single connection, at full capacity
    - Good practice: prefetch data such that you will only need to initiate another download every 2 to 5 minutes, and in the order of 1 to 5 MB
* More efficient to reuse existing network connections than to initiate new ones 🡪 bundle!
* Android **DDMS (Dalvik Debug Monitor Server)** includes a Detailed Network Usage tab that tracks when your application is making network requests
* Every time your app polls your server to check if an update is required, you activate the wireless radio 🡪 waste
  + - **Google Cloud Messaging for Android (GCM)** is lightweight mechanism used to transmit data from a server to a particular app instance
    - This can allow server to notify your app running on a particular device that there is new data available for it
* Where polling is required, it’s good practice to set the default data refresh frequency as low as possible
* Avoid downloading duplicate data
* Modify download patterns based on connectivity type

**Building Apps with User Info & Location**

* **Remembering Users** 
  + Remembering your user using AccountManager
  + Authenticating to OAuth2 Services
* **Making Your App Location Aware** 
  + Can communicate with other devices on the same local network (just have to be physically nearby)
  + You can choose your own Location Provider or let Android decide a close match based on a number of criteria
  + You can set a location listener to receive notifications at least every 10 seconds and if the devices moves by more than 10 meters
  + Location provider with greater accuracy (GPS) requires longer fix time than one with lower accuracy (network-based)
  + Use getLastKnownLocation() wisely
  + Terminate location updates when not in use

**Best Practices for Performance**

* Best Practices for Performances
  + Performance Tips – optimize performance and improve battery efficiency
  + Improving Layout Performance – UI responsiveness
  + Running in a Background Service
  + Loading Data in the Background
  + Optimizing Battery Life
  + Sending Operations to Multiple Threads
  + Keeping Your App Responsive
  + JNI Tips – Using Java Native Interface
  + SMP Primer for Android – Coding on symmetric multiprocessor systems
* Best Practices for Security & Privacy
* Using Google Play to Distribute & Monetize