Author: David Chun
Last Updated: Nov 11th, 2017
Topic: Basic Android Concepts

1. Manifest
    a. Permissions
        i. API 26+ run time
            1. Check permissions in Activity if required
        ii. API < 26 only manifest -> accepts at installation time
    b. Styles
        i. Noactionbar, lightactionbar, darkactionbar, day/night
            1. Actionbar -> header bar used to display menus, titles, drawers, etc.
                a. Set toolbar view as actionbar using setActionBar
                b. Override onCreateOptionsMenu and onOptionsSelectItemSelected to inflate menu and listen to clicks
                c. setDisplayHomeAsUpEnabled shows corner button (can set custom drawable)
    c. Services/providers registration
        i. For more information
    d. Parameters
        i. launchMode (singleTop, singleInstance, etc), theme (from Styles)
2. Activity
    a. Lifecycle
        i. onCreate, onStart, onResume, onPause, onStop
            1. Override to add functionality, call super
            2. Activity Lifecycle Guide
    b. Threads
        i. UI Thread -> don't block or UI will become unresponsive (drops frames)
        ii. Threads vs Asynctask vs Service vs Handler
            1. Background Processing Explanation for more information
            2. Thread -> basic multi-threading tool, pure Java
                a. Need to handle sync, cancellation, pooling, config changes
            3. Asynctask
                a. Encapsulates thread, can report progress, syncs automatically
            4. Service
                a. Decouples execution from context, bind to communicate (interface) -> remember to unbind
            5. Handler
                a. Used to send data to thread registered with (post results)

- c. Layout
  - i. Views
    - 1. Specs
      - a. Layout specs
        - i. Height (minheight), width (minwidth), elevation
        - ii. Background, src, theme
        - iii. Margin vs padding
          - 1. Margin -> outside
          - 2. Padding -> inside
        - iv. Gravity vs layoutgravity
          - 1. Gravity -> children
          - 2. layoutGravity -> view itself
        - v. id -> (@+id/layout_name)
    - 2. Containers
      - a. RelativeLayout
        - i. Relative alignment -> alignStart, alignLeft
      - b. LinearLayout
        - i. Orientation
        - ii. Weight
      - c. CoordinatorLayout
        - i. Link views together
      - d. FrameLayout
        - i. Fills all available space
    - 3. Views
      - a. Options -> same as layouts
      - b. Focus change listener
      - c. <include layout="@layout/layout_name">
      - d. Xlmns tool
      - e. Common views
        - i. Textview, editText, fab, toolbar, imageview, scrollview (nestedscrollview), View, fragment, snackbar, cardview
  - ii. Inflating layouts/menus
    - 1. Menu listener -> access by id
  - iii. Referencing views
    - 1. Requires id parameter
    - 2. findViewById(R.id.layout_name)
    - 3. UI Thread only
      - a. If in different thread, use handler, asynctask, or callbacks
- d. State
  - i. Bundles, SharedPreferences, etc
    - 1. Override onSaveInstanceState to save data, restore in onCreate if Bundle != null

            2. SharedPreferences: mode to set permissions, must open editor and apply changes to commit (can use commit but apply is async)
    e. Instances, new activities, etc
        i. Intents (operation), flags, passing data
            1. Animation -> anim folder resources
            2. Bundles to pass data
        ii. Toasts -> context, string, duration (View.LENGTH_LONG/LENGTH_SHORT)
    f. Context
        i. getContext vs getApplicationContext
            1. Current context vs entire application
    g. Animations (activities/fragments)
        i. Basic -> anim -> override pending animation
    h. Orientation
        i. portrait/landscape
            1. Manifest -> can restrict orientation (screenOrientation) - not rec.
            2. Specific layouts in layout-land folder
            3. Split screen consideration and scalability
        ii. Fullscreen
            1. Window params (sticky)
3. Fragments
    a. Callback/binders(not just fragments)
        i. Interfaces
    b. Lifecycle
        i. onAttach, onCreate, onCreateView, onActivityCreated, onStart, onResume, onPause, onStop, onDestroyView, onDestroy, onDetach
            1. Override to add functionality (eg bind/unbind), call super
        ii. Fragment Lifecycle Guide
    c. Retained fragments -> save state/execution on context change
4. Listeners/adapters
    a. RecyclerViews
        i. Custom views -> custom adapters
    b. Listeners
        i. onClick, stateChanged, custom listeners
5. Resources
    a. Use getResources().getSomething to get resource by id
    b. Strings, Colors, Drawables, Dimen
        i. Strings -> translation purposes, reuse, etc
            1. Arrays/items
        ii. Colors -> day/night, palette
        iii. Drawables -> icon for display density, size vs loading times
        iv. Dimen -> view dimensions
6. Memory/SD Card
    a. Complicated, depends on permissions, intent, etc

b. Uris vs URIs, uri permissions (persist)
   i. Java vs Android
c. DocumentProvider/FileProvider/ContentProvider -> Register in manifest

7. Libraries
   a. Github, Android-Arsenal
      i. Make sure to check Android version and most recent library version
   b. Licenses
      i. Apache, MIT
         1. Attribution Required
      ii. CC
         1. Check type of CC license, some restrict commercial use
      iii. No license -> default copyrighted
8. Builds
   a. Gradle options
      i. Proguard -> obfuscation -> proguard-rules.pro
      ii. API levels -> min API limits API usage
   b. Build apk/aar
      i. Keystore
         1. Keystore for signing apps (V1 and V2)
      ii. Release type
         1. Debug vs Release params
9. Misc
   a. Root -> RootTools (Stericsson or equivalent)
   b. Camera API
   c. ADB Commands