Android VII

Intent Service, Notifications, Pending Intents
**Toy app: Hydration Reminder**

- When finished, the app will remind you to take a drink every 15 min (via a notification) when the phone is charging and keep track of the number of drinks and charges.
- We will be storing number of drinks and charges inside SharedPreferences.
  - SharedPreferences is a very simple persistence class that stores key/value pairs.
- Methods using SharedPreferences will be stored in the PreferenceUtilities class.
Services

- We will use a Service to do the drink and charge count updating
- Data syncing has nothing to do with the UI
  - Don’t use UI components to do it
- Do it in a Service
  - Don’t have a UI component
  - Can run after all the activities are gone
- Services start out on the main thread. Using services (except intent services) requires creating a background thread.
Services versus Loaders

- Services and Loaders kinda’ do the same thing, though in different ways.

- When should we use which?
Quiz: Service or Loader?

- Decoding a Bitmap to be displayed
- Updating the schedule from the internet
- Querying the local database to display in a list
Starting a service

WAYS TO START A SERVICE

Start
Schedule
Bind
Starting a service from an activity

- You can do it directly with `startService()`
- You can use `JobScheduler`
- These are “fire and forget” uses of a Service
**Bound services**

- Services that need to communicate with a UI component (like the music player below) use the bound service pattern.
- It works by calling `bindService` from a UI class.
- `IBinder` objects allow sharing information between services and UI components.
Started Service Lifecycle

- Call to `startService()`
  - `onCreate()`
  - `onStartCommand()`
  - Service Running
  - Start the AsyncTask here.
  - `stopSelf()`
  - `onDestroy()`
  - Service Shut Down

The service is stopped by itself or a client.
**Intent services**

- Automatically runs in background thread
- Starting them is easy
- We must subclass IntentService and put the background thread code in `onHandleIntent`
- We can put data into the bundle of the intent for our service

Note, all intent service requests will be handled in a single background thread.
Toy app coding

In this app we will just use an intent service to update the droplet count every time it is pressed.
Why the Intent Service?

- SharedPreferences updates are quick, but . . .
- What if it could take a long time (e.g., updating news data from an api?)
- Intent services allow it to be run on a background thread
Steps to Implement the IntentService

- Create a new class that extends IntentService
- Override onHandleIntent
- Start the service using startService()
A notification is a message you can display to the user outside of your application's normal UI. When you tell the system to issue a notification, it first appears as an icon in the notification area. To see the details of the notification, the user opens the notification drawer. Both the notification area and the notification drawer are system-controlled areas that the user can view at any time.
Pending intents are intents that can be passed from one component to another to be run later, with all the permissions of the first component.
Toy App: making a notification display

- We will start with notifications by adding a “test notification” button just to see how to make a notification.
- Normally these would be created by our application at appropriate times.