# Presentation Cues

# Section 1:

## Design Principles

* + Who is your target market?
  + Cues and guides
  + Spaces
  + Clean
  + Color, size and shape
  + Inform user (User Experience)

## XML Basics

* XML is a markup language used to format and arrange screen layouts.
* Two formats: One ending in > and one ending in />
* Why do we use xml? Because programmatically typing layouts is a lot of work.
* Main goal for designing layouts: Make things as relative as possible.

## Linear Layout

* Arranges views in a linear fashion
* Don’t be generic, specify orientation
* Best approach is to use weight

## Relative Layout

* Arranges views relative to one another
* Best layout to use with the drag and drop method

## Margins, Padding and Gravity

* Margins is more of extending your layout (Pushes outward): Pushes other layouts away from your content
* Padding is more filling up your layout (Pushes inward): Gives your content more room on the inside
* Gravity: Setting up gravity for the content
* Layout Gravity:

## Adding Images

* Two ways:
  + Adding images in the layout itself: Background
  + Using image view to format the image better. (Using scale type)

## Adding Event Listeners

* .setOnClickListener()

# Section 2:

## Android Lifecycle

* OnCreate(): Incoming Intent -> Setting up UI, Initialization
* OnStart(): Starting, application not visible -> Receives on restart state
* OnRestoreInstanceState(): Restore previous state if needed -> Quickly putting things back together after onStop and on Restart
* OnResume(): Starting, application still not visible -> State that waits for possible user input
* OnPause(): Activity partially visible -> Another intent changing events, another activity taking control : Either goes back to on resume or proceeds to stop
* onSaveInstanceState(): Activity no longer visible -> Save state changes, data either to db or some other temporary wys
* onStop(): Activity no longer visible -> You shutdown code here
* onDestroy(): Application gracefully shutdown : App is not forcefully shut down

## Implicit vs Explicit Intent

* An intent is an abstract description of an operation to be performed.
  + Can be used to start activities, broadcast intents start a service, etc.

## Implicit Intents do not directly specify the Android components which should be called, it only specifies action to be performed. A Uri can be used with the implicit intent to specify data type.

## Intent intent = new Intent(ACTION\_VIEW,Uri.parse("http://www.google.com");

## Android system searches for all components which are registered for the specific action and the data type. If many components are found then the user can select which component to use.

* + - (Intent.ACTION\_VIEW, Uri.parse(<http://facebook.com)>)
    - (Intent.ACTION\_DIAL, Uri.parse(“tel:(+63)9568504890”))
    - (Intent.ACTION\_VIEW, Uri.parse(“geo:30.2715,-32.742”))
    - (“android.media.action.IMAGE\_CAPTURE”))
    - (Intent.ACTION\_VIEW, Uri.parse(“content://contacts/people/1”)
  + Check if intent is safe to use or any activity is available to handle intent

* + - 
  + Allow users to access your app from other apps
    - <intent-filter>  
              <action android:name="android.intent.action.SEND"/>  
              <category android:name="android.intent.category.DEFAULT"/>  
              <data android:mimeType="image/\*"/>  
              <data android:mimeType="text/plain"/>  
          </intent-filter>
* Explicit intents are used in the application itself wherein one activity can switch to other activity.
  + Intent intent = new Intent(this,Target.class);
  + Explicit Intents can also be used to pass data to other activity using putExtra method and retrieved by target activity by getIntent().getExtras() methods.

## Sending Data via Intent

* Including data: Intent -> .putExtra(key, value)
* Waiting for specific result: startActivityForResult(intent,r equestcode) -> onActivityResult(request code, result code, intent data)
* Returning result: setResult(result code, intent data)