Mobile Application Programming: Android Messaging
Activities

- Apps are composed of activities
- Activities are self-contained tasks made up of one screen-full of information
- Activities start one another and are destroyed commonly
- Apps can use activities belonging to another app
Application
Application Controller (MVC)
Model View Adapter (MVA)
Notification

How do these happen? Listeners & Interfaces!
Notification Options

- **Listeners**
  - Declare listener *interface*
  - Add *getter/setter* for listener
  - *Call methods* on listener when appropriate

- **Interfaces** like listeners but more complex (eg. *ListAdapter*)

- **LocalBroadcastManager** (Android Support Package)
  - Call *registerReceiver*(BroadcastReceiver r, IntentFilter i)
  - Send using *sendBroadcast*(Intent i)
Notification Options

- **Listeners**
  - **Pros**: Simple and effective if used with anonymous classes
  - **Cons**: Tedious when many events need to be communicated
  - **Cons**: Only one listener is generally allowed to listen

- **Interfaces** like listeners but more complex (eg. `ListAdapter`)
  - **Pros**: Allows complex interaction
  - **Cons**: Generally means **more code** to write to satisfy large interface

- **LocalBroadcastManager** (Android Support Package)
  - **Pros**: Easy cross-thread messaging, can have many receivers
  - **Cons**: Cannot request data from receiver using a broadcast
Notification Options

- **Listeners**
  - Pros: Simple and effective if used with anonymous classes
  - Cons: Tedious when many events need to be communicated
  - Cons: Only one listener is generally allowed to listen

- **Interfaces** like listeners but more complex (eg. ListAdapter)
  - Pros: Allows complex interaction
  - Cons: Generally means more code to write to satisfy large interface

- **LocalBroadcastManager** (Android Support Package)
  - Pros: Easy cross-thread messaging, can have many receivers
  - Cons: Cannot request data from receiver using a broadcast
Listener Collection

- Create a collection of listeners
- When notifying a single listener, notify all listeners
- Note that this makes asking for information complex because all received data must be considered
- Example: Voting for class president
  - Each voter asked for vote
  - Voter returns preference
  - Accounting of votes determines returned value
Broadcast

- **LocalBroadcastManager**
  - Available in Android Support Package
  - Similar to cross-application broadcast system
  - **Centralized system** to register for notifications
  - Receivers may **begin / end** receiving notifications at any time
  - Broadcaster **can’t ask for information** from receivers!
Broadcast Usage
Messaging Mechanisms

- Listener
- Interface
- Listener Collection
- Broadcast Manager
Broadcast - Example

ListView
ListAdapter
MessagePosted
invalidateViews
addMessage
Interfaces
Methods
Broadcasts
MessageManager

- Broadcast - Example
- ListView
- ListAdapter
- User action
- MessageController
- 1
- addMessage
- MessageManager
- 2
- invalidateViews
- Update
- Notify

- Broadcasts
- Interfaces
- Methods

- MessagePosted
- 3
- 4

- ListView
- ListAdapter
- User action
- MessageController
- 1
- addMessage
- MessageManager
- 2
- invalidateViews
- Update
- Notify

- Broadcasts
- Interfaces
- Methods

- MessagePosted
- 3
- 4

- ListView
- ListAdapter
- User action
- MessageController
- 1
- addMessage
- MessageManager
- 2
- invalidateViews
- Update
- Notify

- Broadcasts
- Interfaces
- Methods

- MessagePosted
- 3
- 4

- ListView
- ListAdapter
- User action
- MessageController
- 1
- addMessage
- MessageManager
- 2
- invalidateViews
- Update
- Notify