SMART DEVICE PROGRAMMING

6133

INTENT

- ✓ An Android application can contain zero or more activities.
- ✓ When your application has more than one activity, you may need to navigate from one activity to another.
- ✓ In Android, you navigate between activities through what is known as an intent.
- ✓ Android application components can connect to other Android applications. This connection is done using an Intent object.
- ✓ Intents allow you to interact with components from the same applications as well as with components contributed by other applications.

- ✓ activities, services, and broadcast receivers—are activated by an asynchronous message called an intent.
- ✓ Intents bind individual components to each other at runtime.
- ✓ You can think of them as the messengers that request an action from other components, whether the component belongs to your app or another.
- ✓ All Android activities are started or activated with an intent.
- ✓ Intents are message objects that make a request to the Android runtime to start an activity or other app component in your app or in some other app.
- ✓ In addition to starting activities, intents are also used to pass data between activities.
- ✓ When you create an intent to start a new activity, you can include information about the data you want that new activity to operate on.

INTENT TYPES

- ✓ There are two types of intents in Android:
- ✓ **Explicit intents** specify the receiving activity (or other component) by that activity's fully-qualified class name. Use an explicit intent to start a component in your own app because you already know the package and class name of that component.
- ✓ **Implicit intents** do not specify a specific activity or other component to receive the intent. Instead you declare a general action to perform in the intent. The Android system matches your request to an activity or other component that can handle your requested action.

INTENT OBJECTS AND FIELDS

- ✓ An Intent object is an instance of the Intent class. For explicit intents, the key fields of an intent include the following:
- ✓ **The activity class** (for explicit intents). This is the class name of the activity or other component that should receive the intent
- ✓ **The intent data.** The intent data field contains a reference to the data you want the receiving activity to operate on, as a Uri (Uniform Resource identifier) object.
- ✓ **Intent extras.** These are key-value pairs that carry information the receiving activity requires to accomplish the requested action.
- ✓ **Intent flags.** These are additional bits of metadata, defined by the Intent class. The flags may instruct the Android system how to launch an activity or how to treat it after it's launched.

STARTING AN ACTIVITY WITH AN EXPLICIT INTENT

- ✓ To start a specific activity from another activity, use an explicit intent and the startActivity() method.
- ✓ Explicit intents include the fully-qualified class name for the activity or other component in the Intent object. All the other intent fields are optional, and null by default.
- ✓ For example, if you wanted to start the ShowMessageActivity to show a specific message in an email app, use code like this.

```
Intent messageIntent = new Intent(this, ShowMessageActivity.class);
startActivity(messageIntent);
```

- ✓ Use the startActivity() method with the new intent object as the only argument.
- ✓ The startActivity() method sends the intent to the Android system, which launches the ShowMessageActivity class on behalf of your app.
- ✓ The new activity appears on the screen, and the originating activity is paused.
- ✓ The started activity remains on the screen until the user taps the back button on the device, at which time that activity closes and is reclaimed by the system, and the originating activity is resumed.
- ✓ You can also manually close the started activity in response to a user action with the finish() method: