

Royal University of Bhutan

Unit II: Activity Intents

CTE308- AS2025

Tutor: Pema Galey
#17682761

Outline

- Activities
- Defining an activity
- Intent
- Starting a new activity with an intent
- Passing data between activities with extras
- Navigating between activities

What is an Activity?

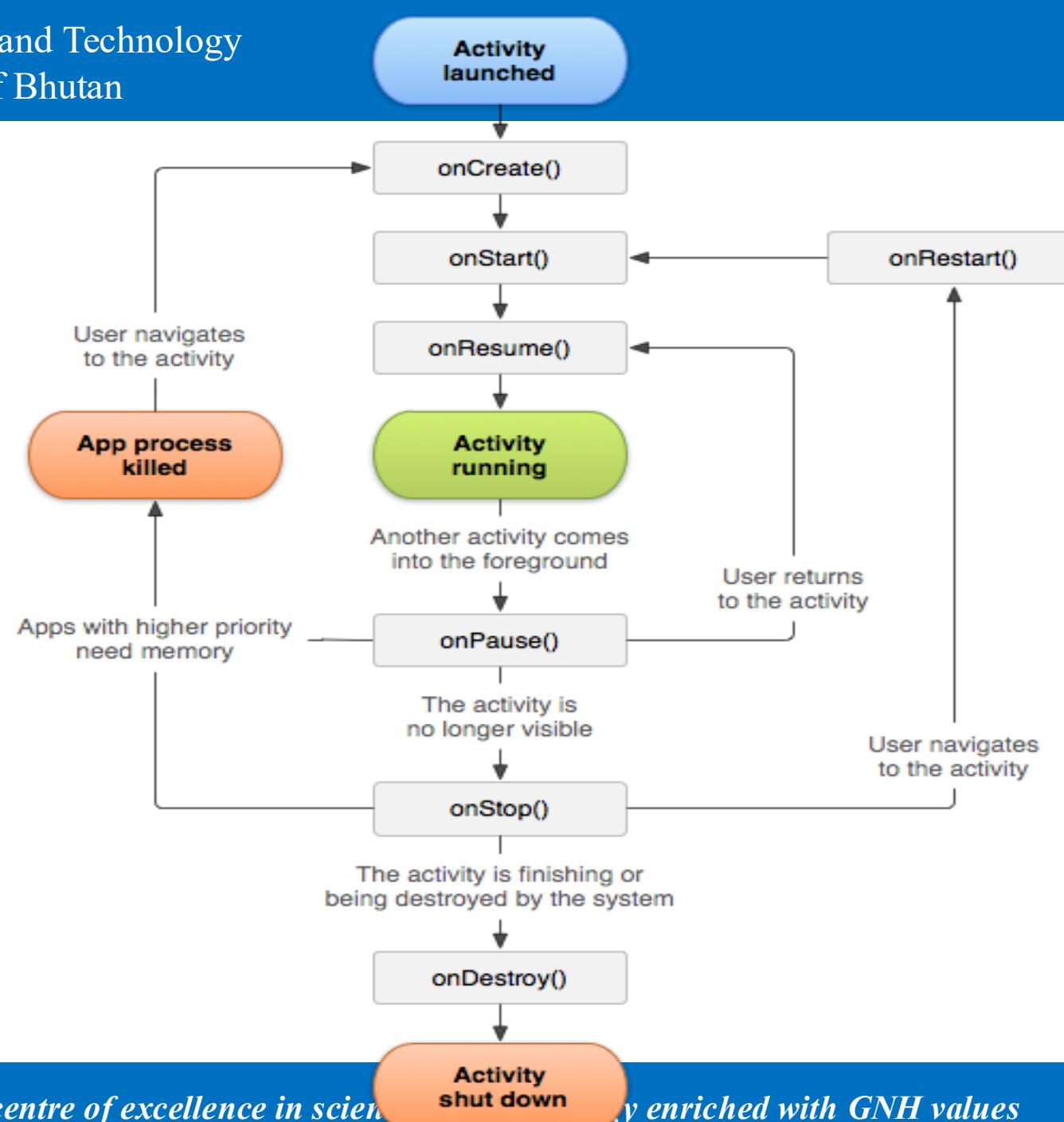
- An Activity is an application component.
- Represents one window, one hierarchy of views.
- Typically fills the screen, but can be embedded in other activity or a appear as floating window.
- Java class, typically one activity in one file.

What does Activity Do?

- Represents an activity, such as ordering groceries, sending email, or getting directions.
- Handles user interactions, such as button clicks, text entry, or login verification.
- Can start other activities in the same or other apps.

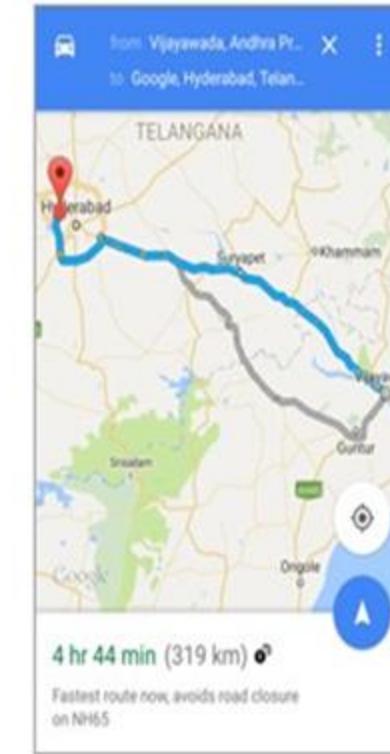
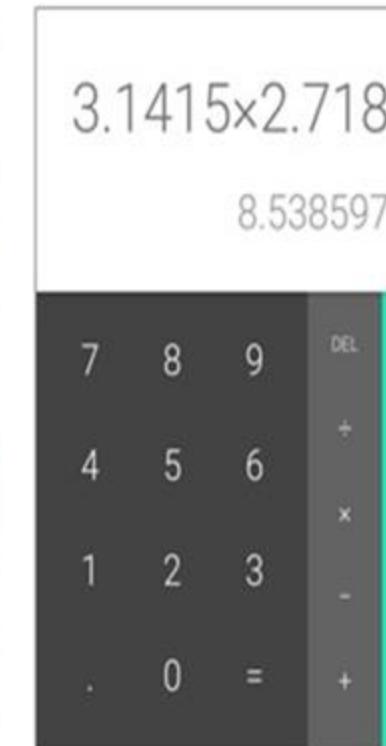
An Activity Lifecycle?

- An Activity has a life cycle.
 - created,
 - started
 - runs,
 - is paused,
 - resumed,
 - stopped, and
 - destroyed.



Examples of Activity

- List of persons, List of different country flags, A calculator, A Map



Apps and Activities

- Activities are **loosely tied** together to make up an app
- First activity user sees is typically called “**MainActivity**”
- Activities can be organized in parent-child relationships in the Android manifest to aid navigation.

Layouts and Activities

- An activity typically has a UI layout.
- Layout is usually defined in one or more XML files.
- Activity "inflates" layout as part of being created.

Implementing Activities

1. Define layout in XML (**activity_main.xml**)

2. Define Activity Java class

- extends AppCompatActivity

```
public class MainActivity extends AppCompatActivity
```

1. Connect Activity with Layout

- Set content view in onCreate()

```
setContentView(R.layout.activity_main);
```

1. Declare Activity in the Android manifest

1. Defining Layout in XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <TextView
        android:id="@+id/textView3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp" />
</LinearLayout>
```

2. Defining Activity Java Class

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
    }  
}
```

3. Connect Activity with Layout

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
}
```

R - Resource

layout - is a layout

activity_main - in this XML file

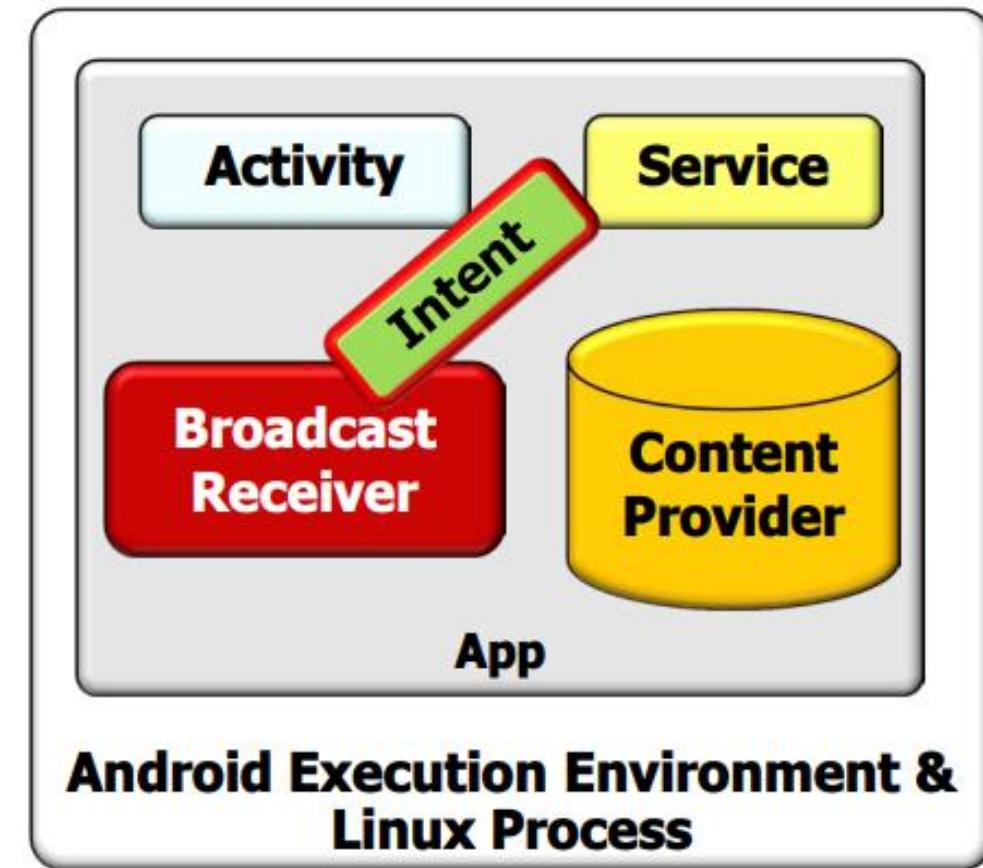
4. Declare Activity in Android Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.myapp">
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Intent

Intent Introduction

- Android uses intents as the “glue” that simplifies the integration of apps that reuse existing components



What is an Intent?

- Android **Intent** lets you navigate from one android activity to another (description of an operation to be performed).
- It can be used to perform following tasks such as :
 - Open another Activity or Service from the current Activity
 - Pass data between Activities and Services
 - Delegate responsibility to another application.
 - For example, you can use Intents to open the browser application to display a URL.

What can Intent do?

— Start activities (Foreground service)

- A button click starts a new activity for text entry.
- Clicking Share opens an app that allows you to post a photo.

— Start services (Background service)

- Initiate downloading a file in the background.

— Deliver broadcasts

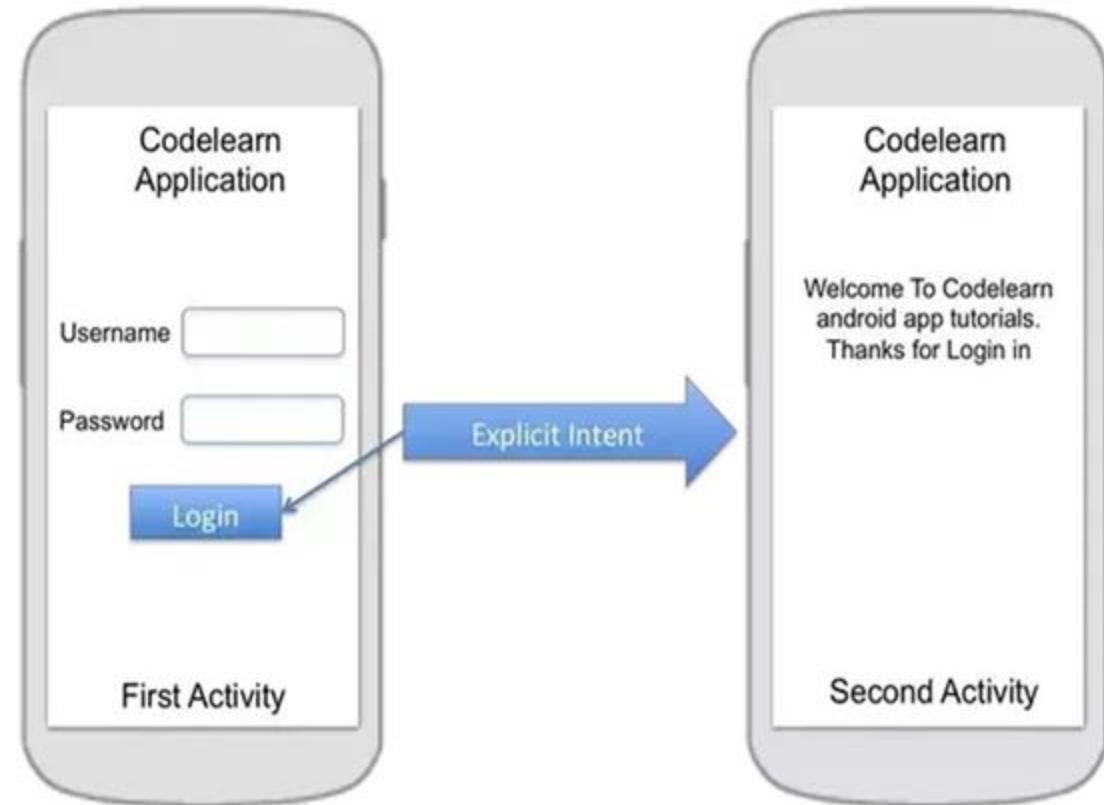
- The system informs everybody that the phone is now charging.

Types of Intent

- Basically, there are two types of Intents.
 1. Explicit Intent
 2. Implicit Intent

1. Explicit Intent

- **Explicit Intent** is the Intent in which you explicitly define the component that needs to be called by **Android System**.
- When you open an activity from another activity in the same Android app, you use explicit Intent.



2. Implicit Intent

- **Implicit Intents** is the intent where instead of defining the components, you define the action you want to perform.
 - The decision to handle this action is left to the operating system.
 - The OS decides which component is best to run for implicit intents.
- Whenever you delegate responsibility to another application from your application, you use Implicit Intents.
- This is typically used when you want to share the data from one application to another.
 - Sharing data over email, sms, social network etc. is a classic example of this category.

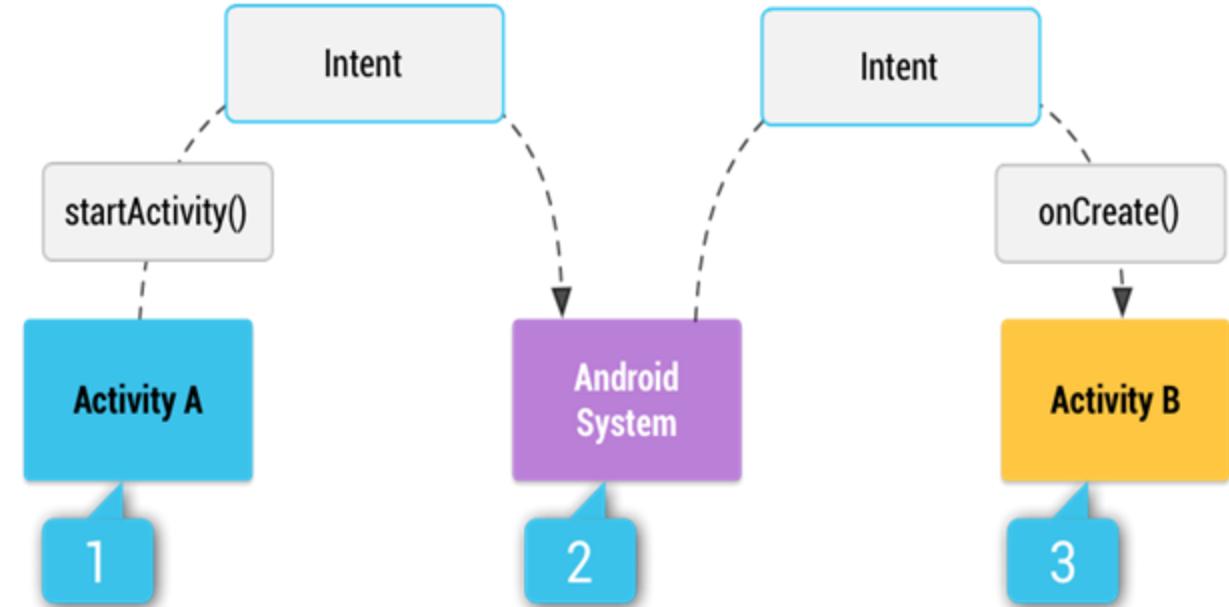
2. Implicit Intent

- Asks system to find an activity that can handle this request
 - Find an open store that sells green tea
 - Clicking Share opens a chooser with a list of apps



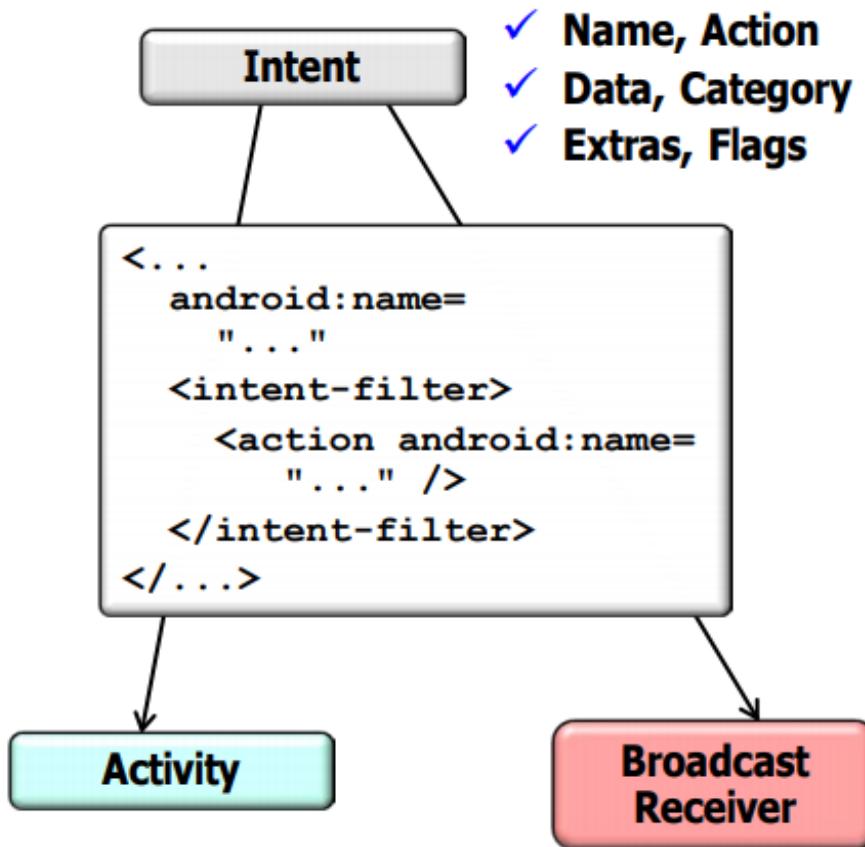
Implicit Intent

- How an implicit intent is delivered through the system to start another activity?
 - [1] Activity A creates an Intent with an action description and passes it to **startActivity()**.
 - [2] The Android System searches all apps for an intent filter that matches the intent. When a match is found,
 - [3] the system starts the matching activity (**Activity B**) by invoking its **onCreate()** method and passing it the Intent.



Implicit Intent

- Implicit intent resolution and intent filter in `AndroidManifest.xml`



Start an Activity with Explicit Intent

- To start a specific activity, use an explicit intent

1. Create an intent

```
Intent intent = new Intent(this, ActivityName.class);
```

1. Use the intent to start the activity

```
startActivity(intent);
```

Start an Activity with Implicit Intent

- To start a specific activity, use an explicit intent

1. Create an intent

```
Intent intent = new Intent(String Action, Uri uri);
```

2. Use the intent to start the activity

```
startActivity(intent);
```

Examples of Implicit Intent

– Loading Web page

```
Uri uri = Uri.parse("http://www.google.com");  
Intent intent = new Intent(Intent.ACTION_VIEW,uri);  
startActivity(intent);
```

– Dial phone number

```
Uri uri = Uri.parse("tel:17000000");  
Intent intent = new Intent(Intent.ACTION_DIAL,uri);  
startActivity(intent);
```

How Activities Run?

- All activities are managed by the Android runtime.
- Started by an "intent", a message to the Android runtime to run an activity.



Sending Data with Intents

- **Data**
 - one piece of information whose data location can be represented by an URI.
- **Extras**
 - one or more pieces of information as a collection of **key-value** pairs in a **Bundle**.

Sending and Retrieving Data

- **In the first (**sending**) activity:**
 - 1) Create the Intent object
 - 2) Put data or extras into that intent
 - 3) Start the new activity with startActivity()
- **In the second (**receiving**) activity:**
 - 1) Get the intent object the activity was started with
 - 2) Retrieve the data or extras from the Intent object

Putting a URI as intent data

- A Web page URI:

```
Uri uri = Uri.parse("http://www.google.com");  
Intent intent = new Intent(Intent.ACTION_VIEW,uri);  
intent.setData(uri);
```

- A Sample File URI:

```
intent.setData(Uri.fromFile(  
        new File("/sdcard/sample.jpg")));
```

Get data from Intents

- **getData();**
- ⇒ Uri locationUri = intent.getData();
- **int getIntExtra (String name, int defaultValue)**
- ⇒ int level = intent.getIntExtra("level", 0);
- **Bundle bundle = intent.getExtras();**
- ⇒ Get all the data at once as a bundle.

Thank you!