

**CSC 472 / 372**  
**Mobile Application**  
**Development for Android**



Prof. Xiaoping Jia  
 School of Computing, CDM  
 DePaul University  
[xjia@cdm.depaul.edu](mailto:xjia@cdm.depaul.edu)  
 @DePaulSWEng

**Outline**

- Android logging system
- Table layout and relative layout
- Check boxes, radio buttons, toggle buttons, switches
- Seek bars, rating bars
- Image views

DEPAUL UNIVERSITY

2

**Android**  
**Logging System**


**Android Logging System**

- The Android logging system provides a mechanism for collecting and viewing debug output from the system, emulator, and your app
- *LogCat* – a utility for viewing and filtering log messages
  - Log messages from the system, emulator, and your apps
  - Available in Android Studio and from the command-line

DEPAUL UNIVERSITY

4

**Use the Log Class**



- The *Log* class defines several static logging methods
  - Print messages viewable in LogCat
- Logging methods, in order of increasing verbosity
  - `Log.e(tag, message)` error, kept in release
  - `Log.w(tag, message)` warning, kept in release
  - `Log.i(tag, message)` information, kept in release
  - `Log.d(tag, message)` debug, debug version only
  - `Log.v(tag, message)` verbose, during development

DEPAUL UNIVERSITY

5

**Logging Conversion**

- The *tag* identifies the source of a log message.
  - i.e., the class or activity where the log call occurs.
- Declare a *TAG* constant in your class
 

```
private static final String TAG = "MainActivity";
```
- Call the log methods
 

```
Log.d(TAG, "A log message");
```

DEPAUL UNIVERSITY

6

## View Log Messages

- Run your App in Debug mode
  - The **Debug** icon on the toolbar
- Open the “Debug” tab
- Click on the “LogCat” tab

Debug

LogCat tab

Debug tab

DEPAUL UNIVERSITY

## View Log Messages

- To filter log messages by log level,
  - select a level under *Log Level*
- To filter log messages with a specific string,
  - enter the string in the search box and press Enter.

Log Level

Search box

Log Level

Search

DEPAUL UNIVERSITY

## Table Layout & Relative Layout

DEPAUL UNIVERSITY

## The Table Layout

- Arranges its children into rows and columns.
  - No border lines.
  - A subclass of *Linear Layout*
- Contains one or more *Table Rows*
  - A subclass of *Linear Layout*
- Each *Table Row* contains one or more cells
  - Cells can be empty.
  - Cells may span multiple columns.

DEPAUL UNIVERSITY

## Table Layout Parameters

- Columns in *Table Layout* can be hidden, stretchable, or shrinkable
- Attributes of the *Table Layout*
  - collapseColumns** – hidden, collapsed
  - stretchColumns** – fill the available space of its parent
  - shrinkColumns** – shrink the columns to fit the table into its parent
- Table Row*'s layout width is fixed to **match\_parent**, and layout height default to **wrap\_content**

DEPAUL UNIVERSITY

## Table Rows and Cells

- Cells in a row are in increasing column order (starting 0)
  - You may specify a column number for a cell
  - Skip a column number, leave an empty cell
- Attributes of the cells
  - layout\_column** – the 0-based index of the column
  - layout\_span** – the number of column to span
- The layout width and height are fixed to **match\_parent** and **wrap\_content**, respectively

DEPAUL UNIVERSITY

### A Login Form App – Using Table Layout

columns

	0	1	2	3
0	User			
1	Password			
2		CANCEL	LOGIN	

rows

DEPAUL UNIVERSITY 16

### The Login Form – Table Layout, Row 0

```
<TableLayout xmlns:android="http://..." android:layout_width="match_parent" android:layout_height="match_parent" android:stretchColumns="1">
<TableRow> ... <TextView ... android:gravity="right"/>
<EditText ... android:layout_span="3"/>
</TableRow>
<TableRow> ... </TableRow>
<TableRow> ... </TableRow>
</TableLayout>
```

Gravity right

Span 3 col.

Stretchable column

DEPAUL UNIVERSITY 17

### The Login Form – Table Layout, Row 1

```
<TableLayout xmlns:android="http://..." android:layout_width="match_parent" android:layout_height="match_parent" android:stretchColumns="1">
<TableRow> ... </TableRow>
<TableRow>
<TextView ... android:gravity="right"/>
<EditText ... android:layout_span="3"/>
</TableRow>
<TableRow> ... </TableRow>
</TableLayout>
```

Gravity right

Span 3 col.

DEPAUL UNIVERSITY 18

### The Login Form – Table Layout, Row 2

```
<TableLayout xmlns:android="http://..." android:layout_width="match_parent" android:layout_height="match_parent" android:stretchColumns="1">
<TableRow> ... </TableRow>
<TableRow> ... </TableRow>
<TableRow>
<Button ... android:layout_column="2"/>
<Button ... />
</TableRow>
</TableLayout>
```

Leave 2 empty cells

DEPAUL UNIVERSITY 19

### The Relative Layout

- A view group that arranges its children based on their relative positions.
- The position of each child view can be relative to
  - its sibling, or
  - the parent
- No need for nested layouts
  - Improved performance

DEPAUL UNIVERSITY 20

### Constraints for Relative Layout

- By default, all children are positioned at the top-left corner of the parent container
- Constraints can be declared in any order
  - $View_1$  is above/below  $View_2$
  - $View_1$  is to the left/right of  $View_2$
  - $View_1$  is aligned with  $View_2$  at left/right/top/bottom/base-line, or start/end based on text flow direction
  - Align the  $View$  with its parent at left/right/top/bottom
  - Center the  $View$  horizontally/vertically/in parent
- No circular dependencies wrt. sizes**

DEPAUL UNIVERSITY 21

## Relative Layout Parameters

- Some layout parameters for *RelativeLayout*

```
layout_alignParentTop    [ true | false ]
Align the top of this child to the top edge of the parent.
```

```
layout_centerVertical   [ true | false ]
Center this child vertically within its parent.
```

```
layout_below             view ID
Positions this child below the designated sibling.
```

```
layout_toRightOf        view ID
Positions this child to the right of the designated sibling.
```

DEPAUL UNIVERSITY 22

## The Message App – Using Relative Layout

To: \_\_\_\_\_  
Subject: \_\_\_\_\_  
Message:  
\_\_\_\_\_

CANCEL SEND

DEPAUL UNIVERSITY 23

## The Message App – Relative Layout, 1<sup>st</sup> Row

```
<RelativeLayout ...
    android:layout_width="match_parent"
    android:layout_height="match_parent" >
    <TextView ...
        android:id="@+id/to_label"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/to" />
    <EditText ...
        android:id="@+id/to"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignStart="@+id/subject" />
    ...
</RelativeLayout>
```

DEPAUL UNIVERSITY 24

## The Message App – Relative Layout, 1<sup>st</sup> Row (Alt)

To: \_\_\_\_\_  
Subject: \_\_\_\_\_  
Message:  
\_\_\_\_\_

CANCEL SEND

DEPAUL UNIVERSITY 25

## The Message App – Relative Layout, 2<sup>nd</sup> Row

```
<RelativeLayout ...
    ...
    <TextView ...
        android:id="@+id/subject_label"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/subject" />
    <EditText ...
        android:id="@+id/subject"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignStart="@+id/subject_label" />
    ...
</RelativeLayout>
```

DEPAUL UNIVERSITY 26

## The Message App – Relative Layout, 3<sup>rd</sup> & 4<sup>th</sup> Row

To: \_\_\_\_\_  
Subject: \_\_\_\_\_  
Message:  
\_\_\_\_\_

CANCEL SEND

DEPAUL UNIVERSITY 27

## The Message App – Relative Layout, 5<sup>th</sup> Row

```
<RelativeLayout ...>
    ...
    <Button
        android:id="@+id/ok"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/entry"
        android:layout_alignParentEnd="true" />
    <Button
        android:id="@+id/cancel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_toStartOf="@+id/ok"
        android:layout_alignTop="@+id/ok" />
</RelativeLayout>
```

To: \_\_\_\_\_  
Subject: \_\_\_\_\_  
Message: \_\_\_\_\_

CANCEL SEND

DEPAUL UNIVERSITY 28

## Checkboxes and Radio Buttons

DEPAUL UNIVERSITY

## Checkboxes and Radio Buttons

- Buttons with a binary state that can be toggled by the user.
  - Boolean attribute: `checked`
- Checkbox**
  - A group of selectable options that are not mutually exclusive.
- Radio button**
  - Similar to checkboxes, except that only one option can be selected in the group.

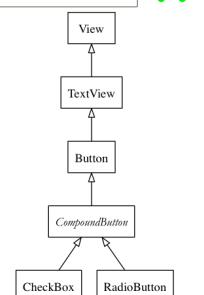
Check Box

Radio Button

DEPAUL UNIVERSITY 30

## Check Boxes

- Class:  
`android.widget.CheckBox`
  - Subclass of *Compound Button* and *Button*
- XML element: `<CheckBox>`
  - All attributes of *Button* are available in *Check Box*
  - Boolean attribute: `checked`



DEPAUL UNIVERSITY 31

## Handle Events of Check Boxes

- Use the `android:onClick` attribute of the `<CheckBox>` element to specify an action method name in the activity class
 

```
public void actionPerformed(View view)
```

  - Invoked when the check box is clicked, i.e., touched
- Set an event listener
  - Interface: `CompoundButton.OnCheckedChangeListener`
  - Action method:
 

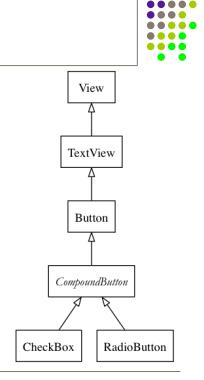
```
public void onCheckedChanged(CompoundButton button,
                                      boolean isChecked)
```

    - Invoked when the check box state is changed

DEPAUL UNIVERSITY 32

## Radio Buttons

- Class:  
`android.widget.RadioButton`
  - Subclass of *Compound Button* and *Button*
- XML element: `<RadioButton>`
  - All attributes of *Button* are available in *Radio Button*
  - Boolean attribute: `checked`
  - Usually placed inside a *Radio Group*



DEPAUL UNIVERSITY 33

## Radio Group

- Defines an exclusive selection scope for a group of *Radio Buttons*
- A subclass of *Linear Layout*
  - Can be horizontal (default) or vertical
- Radio Buttons* must be direct children of a *Radio Group*
  - A radio button that is not a direct child of a radio group will be independent from other radio buttons

DEPAUL UNIVERSITY

34

## Check Box & Radio Button Demo App

- A single screen app
- A group of *Check Boxes*
  - Initial states
- Two separate groups of *Radio Buttons*
  - Horizontal & vertical *Radio Groups*
- Event handling
  - Both approaches



DEPAUL UNIVERSITY

37

## Define the Check Boxes – The Layout

```
<LinearLayout ...>
  ...
  <LinearLayout ...>
    <CheckBox android:id="@+id/checkbox3" ...>
      ...
      <android:onClick="onCheckboxClicked" />
    </CheckBox>
    <CheckBox android:id="@+id/checkbox4" ...>
      ...
      <android:onClick="onCheckboxClicked" />
    </CheckBox>
  </LinearLayout>
</LinearLayout>
```

DEPAUL UNIVERSITY

39

## Handle Events of Radio Buttons

- Use the `android:onClick` attribute of the `<RadioButton>` element to specify an action method name in the activity class
 

```
public void actionPerformed(View view)
```

  - Invoked when the radio button is clicked, i.e., touched
- Set an event listener
  - Interface: `CompoundButton.OnCheckedChangeListener`
  - Action method:
 

```
public void onCheckedChanged(CompoundButton button, boolean isChecked)
```
  - Invoked when the radio button state is changed

DEPAUL UNIVERSITY

36

## Define the Check Boxes – The Layout

```
<LinearLayout ...>
  ...
  <LinearLayout ...>
    <CheckBox android:id="@+id/checkbox1" ...>
      ...
      <android:onClick="onCheckboxClicked" />
    </CheckBox>
    <CheckBox android:id="@+id/checkbox2" ...>
      ...
      <android:onClick="onCheckboxClicked" />
    </CheckBox>
  </LinearLayout>
</LinearLayout>
```



DEPAUL UNIVERSITY

38

## Define the Check Boxes – The Layout

```
<RadioGroup ...>
  ...
  <RadioButton android:id="@+id/radio1" ...>
    ...
    <android:onClick="onRadioButtonClicked" />
  </RadioButton>
  <RadioButton android:id="@+id/radio2" ...>
    ...
    <android:onClick="onRadioButtonClicked" />
  </RadioButton>
  <RadioButton android:id="@+id/radio3" ...>
    ...
    <android:onClick="onRadioButtonClicked" />
  </RadioButton>
</RadioGroup>
```



DEPAUL UNIVERSITY

40

## Define Radio Buttons – The Layout

`<RadioGroup ...>`

```

<RadioGroup ...
    android:orientation="horizontal">
    <RadioButton android:id="@+id/radio4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Single"
        android:onClick="onRadioButtonClicked" />
    <RadioButton android:id="@+id/radio5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Double"
        android:onClick="onRadioButtonClicked" />
    <RadioButton android:id="@+id/radio6"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Suite"
        android:onClick="onRadioButtonClicked" />
</RadioGroup>

```

DEPAUL UNIVERSITY 41

## Handle Actions in the Activity – onClick Actions

The view that triggered the event

```

public void onCheckboxClicked(View view) {
    CheckBox checkBox = (CheckBox) view;
    Log.d(TAG, "onCheckboxClicked() " + checkBox.getText() +
        " " + (checkBox.isChecked() ? "checked" : "unchecked"));
}

public void onRadioButtonClicked(View view) {
    RadioButton radioButton = (RadioButton) view;
    Log.d(TAG, "onRadioButtonClicked() " + radioButton.getText() +
        " " + (radioButton.isChecked() ? "selected" : "unselected"));
}

```

The state of the Check Box

DEPAUL UNIVERSITY 44

## Handle Actions in the Activity – Implement the Action Listener

The listener interface

```

public class MainActivity extends Activity
    implements CompoundButton.OnCheckedChangeListener {
    ...
    @Override
    public void onCheckedChanged(CompoundButton button,
        boolean isChecked) {
        Toast.makeText(this, button.getText() + " is " +
            (isChecked ? "on" : "off"),
        Toast.LENGTH_SHORT).show();
    }
    ...
}

```

The Button that triggered the event and its state

DEPAUL UNIVERSITY 47

## Handle Actions in the Activity – Set Action Listeners

What happens when a radio button is clicked?

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    ...
    CheckBox checkBox1 =
        (CheckBox) findViewById(R.id.checkbox1);
    checkBox1.setOnCheckedChangeListener(this);
    ...
    RadioButton radioButton1 =
        (RadioButton) findViewById(R.id.radio1);
    radioButton1.setOnCheckedChangeListener(this);
}

```

DEPAUL UNIVERSITY 49

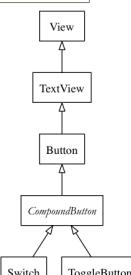
## Toggle Buttons and Switches

- Two-state buttons
- ToggleButton*
  - An on/off button
    - Button + on/off indicator
- Switch* (4.0 or later)
  - 4.0 – 4.4 (API level 14 – 20)
    - Click on, click off
  - 5.0+ (API level 21 or later)
    - Click on, click off
    - Slide/drag action
      - Not the same as a click

DEPAUL UNIVERSITY 51

## Toggle Buttons and Switches

- Class:  
`android.widget.ToggleButton`  
`android.widget.Switch`
  - Subclass of *Compound Button* and *Button*
- XML element:  
`<ToggleButton>`  
`<Switch>`
  - All attributes of *Button* are available in *ToggleButton* and *Switch*
  - Boolean attribute: `checked`



DEPAUL UNIVERSITY 52

## Handle Events of Toggle Buttons & Switches

- Use the `android:onClick` attribute of the `<ToggleButton>` or `<Switch>` element to specify an action method name in the activity class  
`public void actionPerformed(View view)`  

- Invoked when the toggle button or switch is clicked
- Set an event listener
  - Interface: `CompoundButton.OnCheckedChangeListener`
  - Action method:  
`public void onCheckedChanged(CompoundButton button, boolean isChecked)`
- Invoked when the toggle button or switch state is changed

DEPAUL UNIVERSITY 54

## Toggle Button & Switch Demo

- A single screen app
- Toggle Buttons*
  - Standard look
  - Customized text for on and off states
- Switches*
  - Standard look
  - Customized text for on and off states



DEPAUL UNIVERSITY 55

## Toggle Button & Switch Demo

- In KitKat (4.x)
  - The switches have a different look



DEPAUL UNIVERSITY 56

## Define Toggle Buttons – Standard Look

```

<LinearLayout ... >
<ToggleButton
    android:id="@+id/toggle1"
    android:tag="toggle1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="onToggleClicked" />
<ToggleButton
    android:id="@+id/toggle2"
    android:tag="toggle2"
    android:checked="true"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="onToggleClicked" />
</LinearLayout>
  
```

The tag attribute  
"On" state

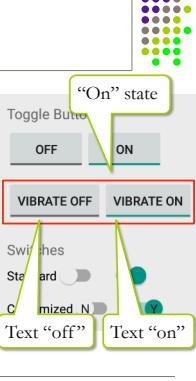


DEPAUL UNIVERSITY 59

## Define Toggle Buttons – Customized Text

```

<ToggleButton
    android:id="@+id/toggle3"
    android:tag="toggle3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textOn="Vibrate on"
    android:textOff="Vibrate off"
    android:onClick="onToggleClicked" />
<ToggleButton
    android:id="@+id/toggle4"
    android:tag="toggle4"
    android:checked="true"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textOn="Vibrate on"
    android:textOff="Vibrate off"
    android:onClick="onToggleClicked" />
  
```

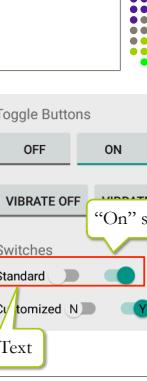


DEPAUL UNIVERSITY 62

## Define Switches – Standard Look

```

<Switch
    android:id="@+id/switch1"
    android:tag="switch1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp"
    android:text="Standard"
    android:onClick="onSwitchClicked" />
<Switch
    android:id="@+id/switch2"
    android:tag="switch2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="10dp"
    android:checked="true"
    android:onClick="onSwitchClicked" />
  
```



DEPAUL UNIVERSITY 65

## Define Switches – Customized Text

```
<Switch
    android:id="@+id/switch3"
    android:tag="switch3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Customized"
    android:textOn="Y"
    android:textOff="N"
    android:onClick="onSwitchClicked" />
<Switch
    android:id="@+id/switch4"
    android:tag="switch4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingLeft="10dp"
    android:checked="true"
    android:textOn="Y"
    android:textOff="N"
    android:onClick="onSwitchClicked" />
```

**Toggle Buttons**

68 DEPAUL UNIVERSITY

## Handle Actions in the Activity – onClick Actions

The view that triggered the event

Use the tag attribute

The state of the Toggle Button

```
public void onToggleClicked(View view) {
    ToggleButton toggleButton = (ToggleButton) view;
    Log.d(TAG, "onToggleClicked() " + toggleButton.getTag() +
        " " + (toggleButton.isChecked() ? "on" : "off"));
}

public void onSwitchClicked(View view) {
    Switch sw = (Switch) view;
    Log.d(TAG, "onSwitchClicked() " + sw.getTag() +
        " " + (sw.isChecked() ? "on" : "off"));
}
```

72 DEPAUL UNIVERSITY

## Handle Actions in the Activity – Implement the Action Listener

```
public class MainActivity extends Activity
    implements CompoundButton.OnCheckedChangeListener {
    ...
    @Override
    public void onCheckedChanged(CompoundButton button,
        boolean isChecked) {
        Toast.makeText(this, button.getTag() + " is " +
            (isChecked ? "on" : "off"),
        Toast.LENGTH_SHORT).show();
    }
}
```

The listener interface

The Button that triggered the event and its state

Use the tag attribute

76 DEPAUL UNIVERSITY

## Handle Actions in the Activity – Set Action Listeners

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    ...
    ToggleButton toggleButton1 =
        (ToggleButton) findViewById(R.id.toggle1);
    toggleButton1.setOnCheckedChangeListener(this);
    ...
    Switch switch1 =
        (Switch) findViewById(R.id.switch1);
    switch1.setOnCheckedChangeListener(this);
}
```

77 DEPAUL UNIVERSITY

## Seek Bars and Rating Bars

## Seek Bars and Rating Bars

- Seek Bar**
  - Interactive slider for selecting a value from a continuous or discrete range of values by moving the slider thumb.
- Rating Bar**
  - Shows a rating in stars.
  - Can set the rating by touch or drag

```

graph TD
    View[View] --> ProgressBar[ProgressBar]
    ProgressBar --> AbsSeekBar[AbsSeekBar]
    AbsSeekBar --> RatingBar[RatingBar]
    AbsSeekBar --> SeekBar[SeekBar]

```

79 DEPAUL UNIVERSITY

## Seek Bars

- Class  
`android.widget.SeekBar`
- XML element  
`<SeekBar>`
- Attributes
  - `max` – The maximum value (minimum is always 0)
  - `progress` – The current value, between 0 and max

DEPAUL UNIVERSITY

80

## Handle Seek Bar Events

- Use an event listener
- Interface: `SeekBar.OnSeekBarChangeListener`
- Action methods:
 

```
void onProgressChanged(SeekBar seekBar,
                      int progress,
                      boolean fromUser)
```

The progress level has changed.

`void onStartTrackingTouch(SeekBar seekBar)`

The user has started a touch gesture.

`void onStopTrackingTouch(SeekBar seekBar)`

The user has finished a touch gesture.

DEPAUL UNIVERSITY

82

## Rating Bars

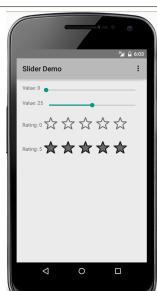
- Class  
`android.widget.RatingBar`
- XML element  
`<RatingBar>`
- Attributes
  - `numStars` – The number of stars
  - `rating` – The current rating, between 0 and numStars
  - `stepSize` – The step size of the rating

DEPAUL UNIVERSITY

81

## The Slider Demo App

- A single screen app
- Seek Bars
  - Default setting
  - Customized setting
- Rating Bars
  - Default setting
  - Customized setting

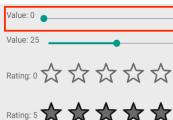


DEPAUL UNIVERSITY

84

## Define Seek Bar – Default Setting

```
<LinearLayout ...>
<TextView
    android:id="@+id/value1"
    android:text="Value: 0"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
<SeekBar
    android:id="@+id/seekbar1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
</LinearLayout>
```



DEPAUL UNIVERSITY

85

### Define Seek Bar – Customized Setting

```
<LinearLayout ...>
    <TextView
        android:id="@+id/value2"
        android:text="Value: 25"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <SeekBar
        android:id="@+id/seekbar2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:max="50"
        android:progress="25" />
</LinearLayout>
```

Value: 0      Value: 25      Rating: 0 ★ ★ ★ ★ ★  
Rating: 5 ★ ★ ★ ★ ★

DEPAUL UNIVERSITY 86

### Define Rating Bar – Default Setting

```
<LinearLayout ...>
    <TextView
        android:id="@+id/value3"
        android:text="Rating: 0"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_vertical"/>
    <RatingBar
        android:id="@+id/ratingkbar1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:numStars="5" />
</LinearLayout>
```

Value: 0      Value: 25      Rating: 0 ★ ★ ★ ★ ★  
Rating: 5 ★ ★ ★ ★ ★

DEPAUL UNIVERSITY 87

### Define Rating Bar – Customized Setting

```
<LinearLayout ...>
    <TextView
        android:id="@+id/value4"
        android:text="Rating: 5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_vertical"/>
    <RatingBar
        android:id="@+id/ratingkbar2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:numStars="5"
        android:rating="5"
        android:stepSize="0.5" />
</LinearLayout>
```

Value: 0      Value: 25      Rating: 0 ★ ★ ★ ★ ★  
Rating: 5 ★ ★ ★ ★ ★

DEPAUL UNIVERSITY 88

### Handle Actions in the Activity – Implement the Seek Bar Listener

```
protected void onCreate(Bundle savedInstanceState) {
    ...
    final TextView value1 = (TextView) findViewById(R.id.value1);
    SeekBar seekBar1 = (SeekBar) findViewById(R.id.seekbar1);
    seekBar1.setOnSeekBarChangeListener(
        new SeekBar.OnSeekBarChangeListener() {
            public void onProgressChanged(SeekBar seekBar,
                int i, boolean b) {
                Log.d(TAG, "onProgressChanged");
                value1.setText("Value: " + i);
            }
            public void onStartTrackingTouch(SeekBar seekBar) { }
            public void onStopTrackingTouch(SeekBar seekBar) { }
        });
    ...
}
```

The listener

Value: 0      Value: 25      Rating: 0 ★ ★ ★ ★ ★  
Rating: 5 ★ ★ ★ ★ ★

DEPAUL UNIVERSITY 89

### Handle Actions in the Activity – Implement the Seek Bar Listener

```
protected void onCreate(Bundle savedInstanceState) {
    ...
    final TextView value2 = (TextView) findViewById(R.id.value2);
    SeekBar seekBar2 = (SeekBar) findViewById(R.id.seekbar2);
    seekBar2.setOnSeekBarChangeListener(
        new SeekBar.OnSeekBarChangeListener() {
            public void onProgressChanged(SeekBar seekBar,
                int i, boolean b) {
                Log.d(TAG, "onProgressChanged");
                value2.setText("Value: " + i);
            }
            public void onStartTrackingTouch(SeekBar seekBar) { }
            public void onStopTrackingTouch(SeekBar seekBar) { }
        });
    ...
}
```

Value: 0      Value: 25      Rating: 0 ★ ★ ★ ★ ★  
Rating: 5 ★ ★ ★ ★ ★

DEPAUL UNIVERSITY 90

### Handle Actions in the Activity – Implement the Rating Bar Listener

```
protected void onCreate(Bundle savedInstanceState) {
    ...
    final TextView value3 = (TextView) findViewById(R.id.value3);
    RatingBar ratingBar1 = (RatingBar) findViewById(R.id.ratingkbar1);
    ratingBar1.setOnRatingBarChangeListener(
        new RatingBar.OnRatingBarChangeListener() {
            public void onRatingChanged(RatingBar ratingBar,
                float v, boolean b) {
                Log.d(TAG, "onRatingChanged");
                value3.setText("Rating: " + v);
            }
        });
    ...
}
```

Value: 0      Value: 25      Rating: 0 ★ ★ ★ ★ ★  
Rating: 5 ★ ★ ★ ★ ★

DEPAUL UNIVERSITY 91

## Handle Actions in the Activity – Implement the Rating Bar Listener

```
protected void onCreate(Bundle savedInstanceState) {
    ...
    final TextView value4 = (TextView) findViewById(R.id.value4);
    RatingBar ratingBar2 = (RatingBar) findViewById(R.id.ratingbar2);
    ratingBar2.setOnRatingBarChangeListener(
        new RatingBar.OnRatingBarChangeListener() {
            public void onRatingChanged(RatingBar ratingBar,
                float v, boolean b) {
                Log.d(TAG, "onRatingChanged");
                value4.setText("Rating: " + v);
            }
        });
    ...
}
```

DEPAUL UNIVERSITY

92

## Image Views

### Image View

- Display a static image
- Can be from various sources, such as drawable resources (including bitmap images)
- Provide many display options

DEPAUL UNIVERSITY

94

### Image View

- Class: `android.widget.ImageView`
- XML element: `<ImageView>`
- Attributes
  - `src` – the source of the image, a drawable resource
  - `scaleType` – controls how the image is resized to match the dimension of the view

DEPAUL UNIVERSITY

95

### Image View Scale Types

- **CENTER**
  - Center the image in the view, no scaling.
- **CENTER\_CROP**
  - Scale the image uniformly (maintain the aspect ratio), and
  - width of the image  $\geq$  width of the view, and
  - height of the image  $\geq$  height of the view
- **CENTER\_INSIDE**
  - Scale the image uniformly (maintain the aspect ratio), and
  - width of the image  $\leq$  width of the view, and
  - height of the image  $\leq$  height of the view

DEPAUL UNIVERSITY

96

### Image View Scale Types

- **FIT\_CENTER, FILL\_START, FILL\_END**
  - Scale the image uniformly (maintain the aspect ratio), and
  - Image fits entirely inside the view, and
  - At least one dimension (width or height) is an exact fit
  - Alignment: center, start, end
- **FIT\_XY**
  - Scale in X and Y independently
  - The image matches both dimensions of the view exactly
  - May change the aspect ratio.

DEPAUL UNIVERSITY

97

## Image View Demo App

- A single screen app that demonstrates various options for displaying images
- Defines several layout files in **res/layout**
  - activity\_main\_demo1.xml, activity\_main\_demo2.xml, ...**
- Change the following line in **MainActivity.java** to load different layout files

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main_demo1);
```

DEPAUL UNIVERSITY 98

## Image View Demo App

- Image source:
  - two bitmap images placed in **res/drawable**
  - android.png** (48 x 48)
  - image001.jpg** (1024 x 768)



DEPAUL UNIVERSITY 99

## Hello Image View! – activity\_main\_demo1.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://..." 
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/image001" />
</LinearLayout>
```



DEPAUL UNIVERSITY 101

## Center, No Scaling – activity\_main\_demo2.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://..." 
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:scaleType="center"
        android:src="@drawable/image001" />
</LinearLayout>
```



DEPAUL UNIVERSITY 103

## Scroll a Large Image – activity\_main\_demo3.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
    xmlns:android="http://..."
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <HorizontalScrollView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:src="@drawable/image001" />
    </HorizontalScrollView>
</ScrollView>
```



DEPAUL UNIVERSITY 104

## Scroll Views

- ScrollView* and *Horizontal ScrollView*
  - Both are subclasses of *View Group*
  - Each holds a *single* child
- Allow the child view to be larger than the parent
- ScrollView* only scrolls vertically
- Horizontal ScrollView* only scrolls horizontally
- Important:** Do not use *ScrollView* with *Text View* or *List View*
  - Will conflict with built-in behavior

DEPAUL UNIVERSITY 105

### Scale a Large Image – activity\_main\_demo4.xml

- Original image 1024 x 768
- Scale to 300dp x 300dp
- Scale type: `default`

```
<ImageView
    android:layout_width="300dp"
    android:layout_height="300dp"
    android:src="@drawable/image001"
    android:background="@color/yellow50"/>
```

DEPAUL UNIVERSITY 106

### Scale a Large Image – activity\_main\_demo4.xml

- Original image 1024 x 768
- Scale to 300dp x 300dp
- Scale type: `center`

```
<ImageView
    android:layout_width="300dp"
    android:layout_height="300dp"
    android:src="@drawable/image001"
    android:scaleType="center"
    android:background="@color/yellow50"/>
```

DEPAUL UNIVERSITY 107

### Scale a Large Image – activity\_main\_demo4.xml

- Original image 1024 x 768
- Scale to 300dp x 300dp
- Scale type: `centerCrop`

```
<ImageView
    android:layout_width="300dp"
    android:layout_height="300dp"
    android:src="@drawable/image001"
    android:scaleType="centerCrop"
    android:background="@color/yellow50"/>
```

DEPAUL UNIVERSITY 108

### Scale a Large Image – activity\_main\_demo4.xml

- Original image 1024 x 768
- Scale to 300dp x 300dp
- Scale type: `centerInside`

```
<ImageView
    android:layout_width="300dp"
    android:layout_height="300dp"
    android:src="@drawable/image001"
    android:scaleType="centerInside"
    android:background="@color/yellow50"/>
```

DEPAUL UNIVERSITY 109

### Scale a Large Image – activity\_main\_demo4.xml

- Original image 1024 x 768
- Scale to 300dp x 300dp
- Scale type: `fitCenter`

```
<ImageView
    android:layout_width="300dp"
    android:layout_height="300dp"
    android:src="@drawable/image001"
    android:scaleType="fitCenter"
    android:background="@color/yellow50"/>
```

DEPAUL UNIVERSITY 110

### Scale a Large Image – activity\_main\_demo4.xml

- Original image 1024 x 768
- Scale to 300dp x 300dp
- Scale type: `fitStart`

```
<ImageView
    android:layout_width="300dp"
    android:layout_height="300dp"
    android:src="@drawable/image001"
    android:scaleType="fitStart"
    android:background="@color/yellow50"/>
```

DEPAUL UNIVERSITY 111

### Scale a Large Image

#### – activity\_main\_demo4.xml

- Original image 1024 x 768
- Scale to 300dp x 300dp
- Scale type: **fitEnd**



```
<ImageView
    android:layout_width="300dp"
    android:layout_height="300dp"
    android:src="@drawable/image001"
    android:scaleType="fitEnd"
    android:background="@color/yellow50"/>
```

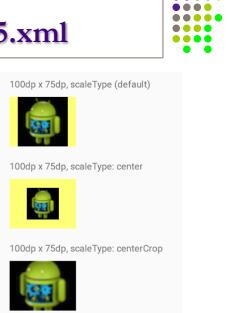
DEPAUL UNIVERSITY

112

### Scale a Small Image

#### – activity\_main\_demo5.xml

- Original image 48 x 48
- Scale to 100dp x 75dp
- Scale type:
  - *default*
  - **center**
  - **centerCrop**



DEPAUL UNIVERSITY

114

### Scale a Large Image

#### – activity\_main\_demo4.xml

- Original image 1024 x 768
- Scale to 300dp x 300dp
- Scale type: **fitXY**



```
<ImageView
    android:layout_width="300dp"
    android:layout_height="300dp"
    android:src="@drawable/image001"
    android:scaleType="fitXY"
    android:background="@color/yellow50"/>
```

DEPAUL UNIVERSITY

113

### Scale a Small Image

#### – activity\_main\_demo5.xml

- Original image 48 x 48
- Scale to 100dp x 75dp
- Scale type:
  - **fitXY**



DEPAUL UNIVERSITY

116

### The Sample Code

- The sample apps in this lecture are available in D2L
  - [Login-TableLayout.zip](#)
  - [Message-RelativeLayout.zip](#)
  - [CheckBoxDemo.zip](#)
  - [ToggleButtonDemo.zip](#)
  - [SlidersDemo.zip](#)
  - [ImageViewDemo.zip](#)
- Each zip archive contains the entire project folder
- Unzip the file and **import** to Android Studio

DEPAUL UNIVERSITY

117

**Next ...**



- Intent and intent filters
- Apps with multiple activities
- Activity lifecycles

◊ Android is a trademark of Google Inc.  
◊ Some contents and diagrams are from Google Inc.  
Licensed under Apache 2.0 and Creative Commons Attribution 2.5.

---

 DEPAUL UNIVERSITY