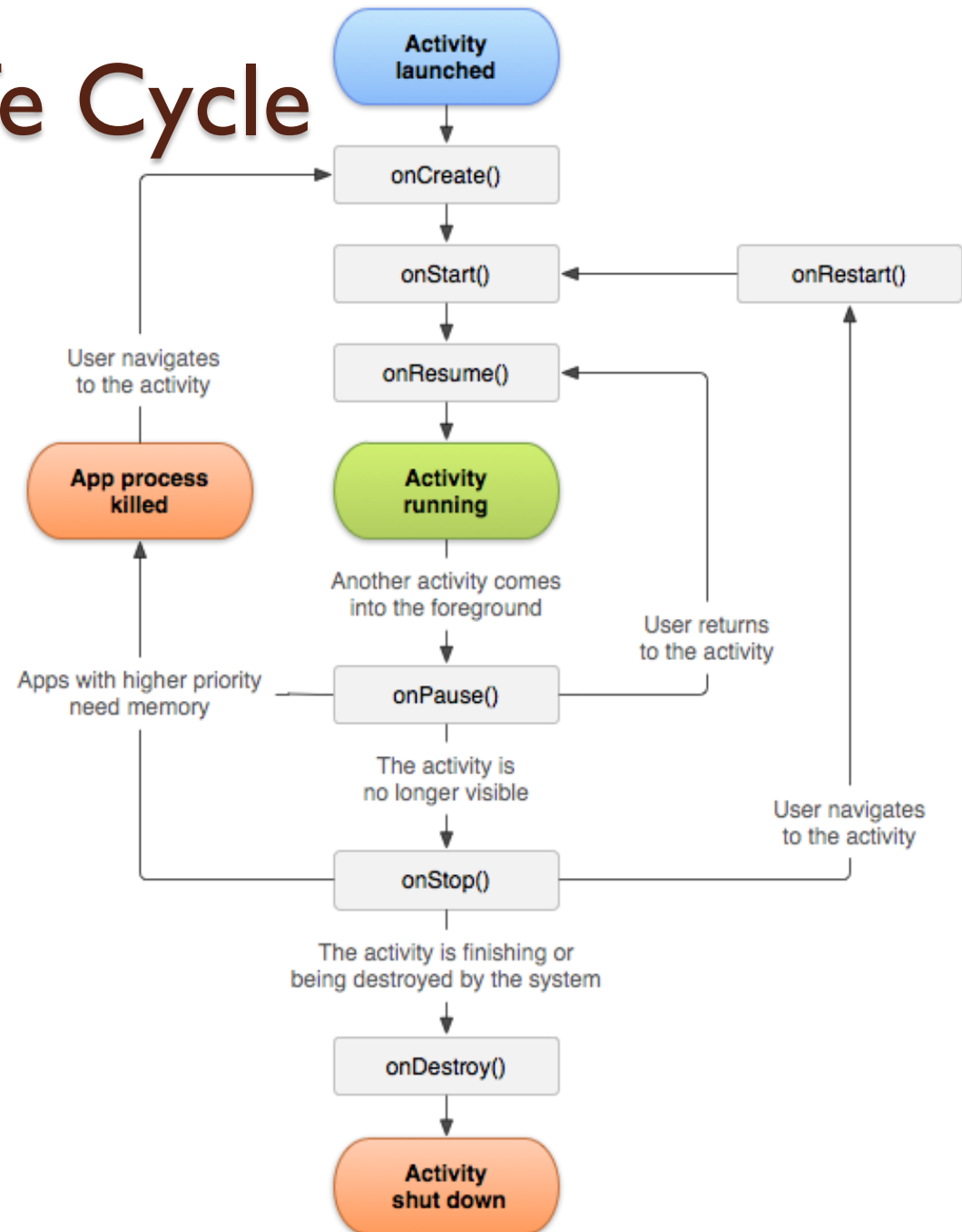# Android Programing

Gavrilut Dragos
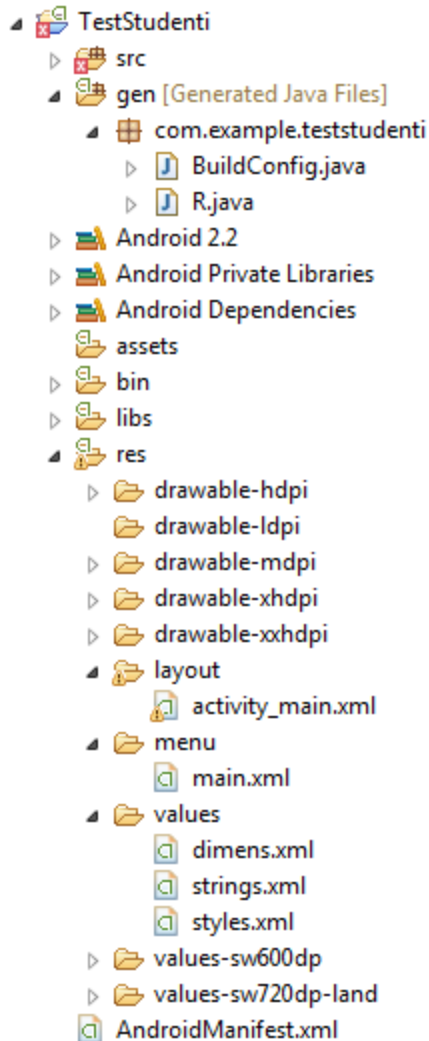
Activity Life Cycle

Activities - onCreate

```java
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main_activity);
}
```

- savedInstanceState – if different than null it represents a object with saved informations about the last state of the activity.
- It is set by **onSaveInstanceState** (Bundle outState)
- setContentView – sets the view object that will be displayed in the current activity
  - **public void** setContentView (**int** layoutResID)
  - **public void** setContentView (**View** view)
  - **public void** setContentView (**View** view, ViewGroup.LayoutParams params)

"R" file


```java
package com.example.teststudenti;

public final class R {
    public static final class attr {
    }
    public static final class dimen {
        public static final int activity_horizontal_margin=0x7f040000;
        public static final int activity_vertical_margin=0x7f040001;
    }
    public static final class drawable {
        public static final int ic_launcher=0x7f020000;
    }
    public static final class id {
        public static final int action_settings=0x7f080004;
        public static final int button1=0x7f080000;
        public static final int button2=0x7f080002;
        public static final int button3=0x7f080003;
        public static final int button4=0x7f080001;
    }
    public static final class layout {
        public static final int activity_main=0x7f030000;
    }
    public static final class menu {
        public static final int main=0x7f070000;
    }
    public static final class string {
        public static final int action_settings=0x7f050001;
        public static final int app_name=0x7f050000;
        public static final int hello_world=0x7f050002;
    }
    public static final class style {
        public static final int AppBaseTheme=0x7f060000;
        public static final int AppTheme=0x7f060001;
    }
}
```

View

- The superclass for every UI design object in Android OS
- The following functions can be overwritten in a view

Function	Action
onMeasure(int, int)	Called to compute the size of the view
onLayout(boolean, int, int, int, int)	Called to compute its layout position
onSizeChanged(int, int, int, int)	Called when the size of the view changed
onDraw(android.graphics.Canvas)	Called to draw the content of the view
onKeyDown(int, KeyEvent)	Called when a key is presses
onKeyUp(int, KeyEvent)	Called when a key is release
onTrackballEvent(MotionEvent)	Called when a track ball event occurs
onTouchEvent(MotionEvent)	Calles when a touch event occurs

View

- UI Objects derived from view
  - ImageView
    - ImageButton
      - ZoomButton
  - ProgressBar
  - SurfaceView
  - TextView
    - Button
      - CheckBox
      - RadioBox
      - ToggleButton

View

- A view and its children can be design using 2 methods: XML design or direct code
- Using XML design means using different layouts to arrange the UI objects in a View.

Layout object	API	Description
AbsoluteLayout	1	Arrange its children after their absolute position (x,y). **Deprecated since API Level 3**
FrameLayout	1	Displays one child (in a frame)
LinearLayout	1	Arrange its childern in one row or one column
RelativeLayout	1	Arrange its children in relation one to another
TableLayout	1	Arrange its childern in a Table (similar to a table in a HTML view)
GridLayout	14	Arrange its children in a Grid (with columns and rows)

View layout XML parameters

- android:layout_width
- android:layout_height
  - fill_parent → as big as its parent (minus the padding). Deprecated since API Level 8 and replaced with math_parent
  - match_parent
  - wrap_content → as big as its childs require plus the padding. In case of Buttons, TextView,… it is as big as the text it contains.
  - or a unit (in pixels (px), density independent pixels (dp), inches (in) , millimeters (mm)

View layout XML parameters

- android:id → object id (will be used in <u>findViewById</u> function).
  Exemple: android:id="**@+id**/<u>button2</u>"

- android:enabled → True/False

- android:onClick → name of the function to process the onClick event for the View

```
@Override

public void onCreate(Bundle savedInstanceState)

{

    super.onCreate(savedInstanceState);

    …

}

public void OnButton1Click(View button)

{

    …

}
```

```
<Button
    android:id="@+id/button1"
    android:onClick="OnButton1Click"
    android:text="Button" />
```
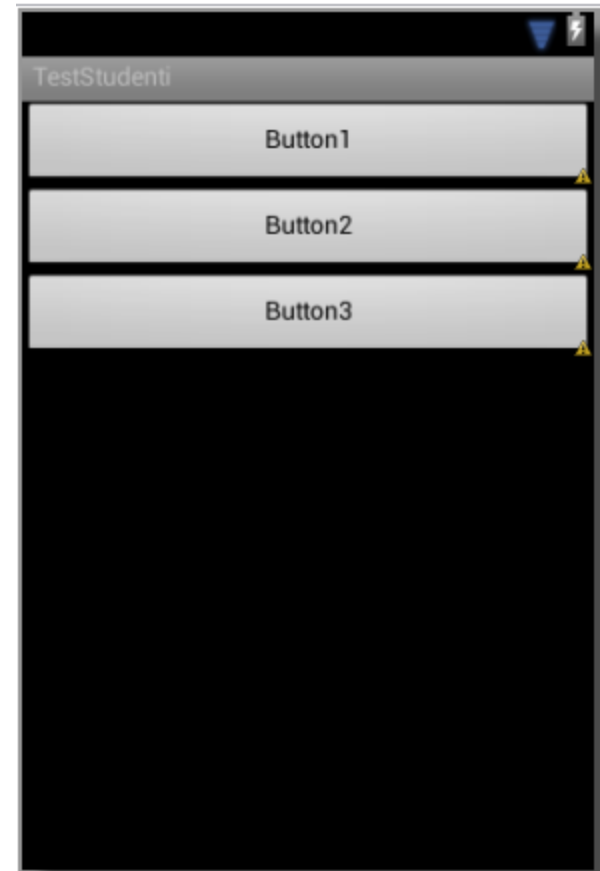
Linear Layout

```xml
<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Button1" />
    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Button2" />
    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Button3" />

</LinearLayout>
```
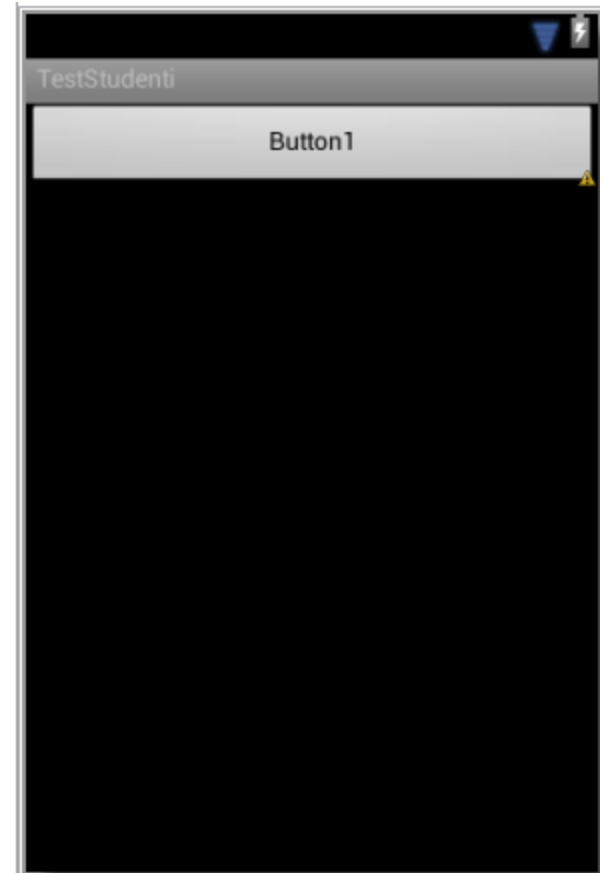
Linear Layout

```xml
<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="horizontal" >

    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Button1" />
    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Button2" />
    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Button3" />

</LinearLayout>
```
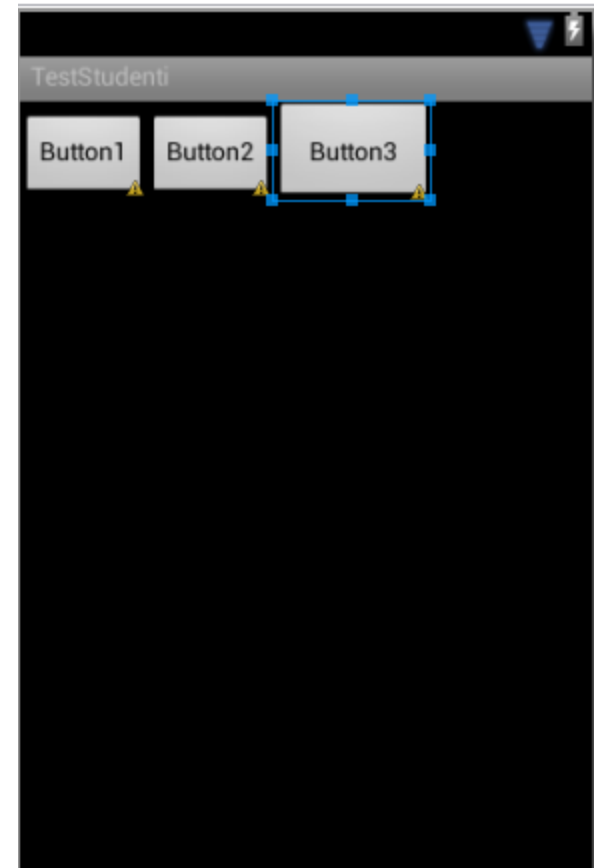
Linear Layout

```xml
<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="horizontal" >

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button1" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button2" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button3" />

</LinearLayout>
```

Linear Layout

```xml
<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal" >
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button1" />
        <Button
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Button2" />
    </LinearLayout>
    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Button3" />
</LinearLayout>
```
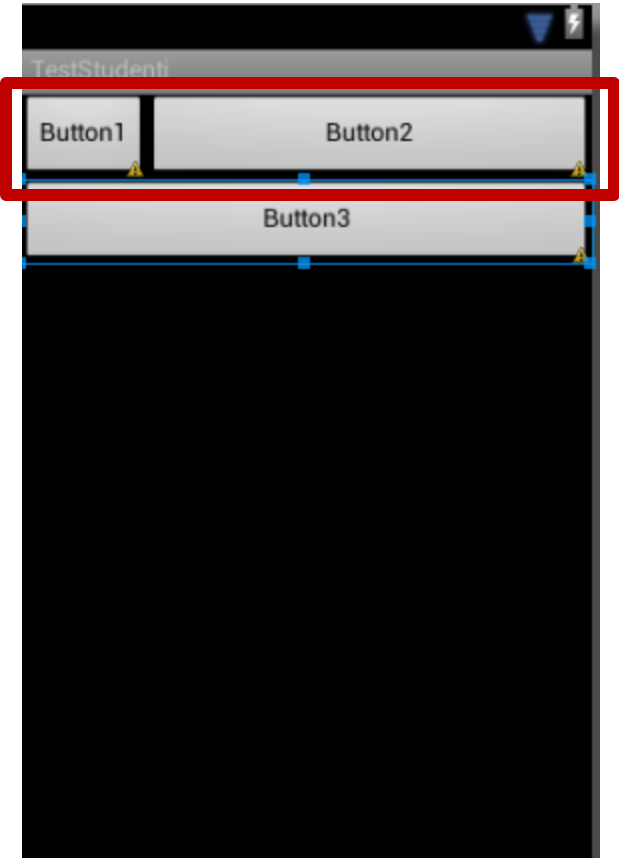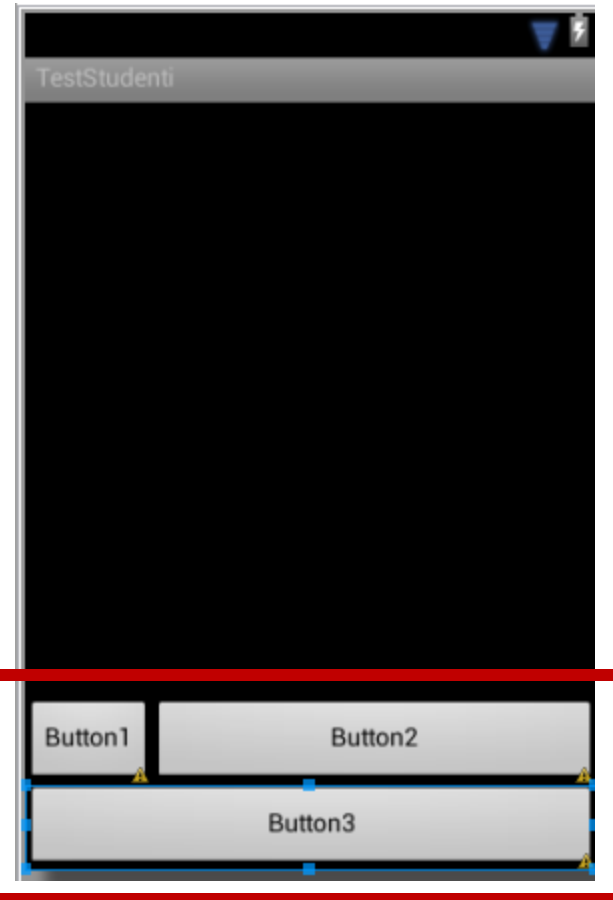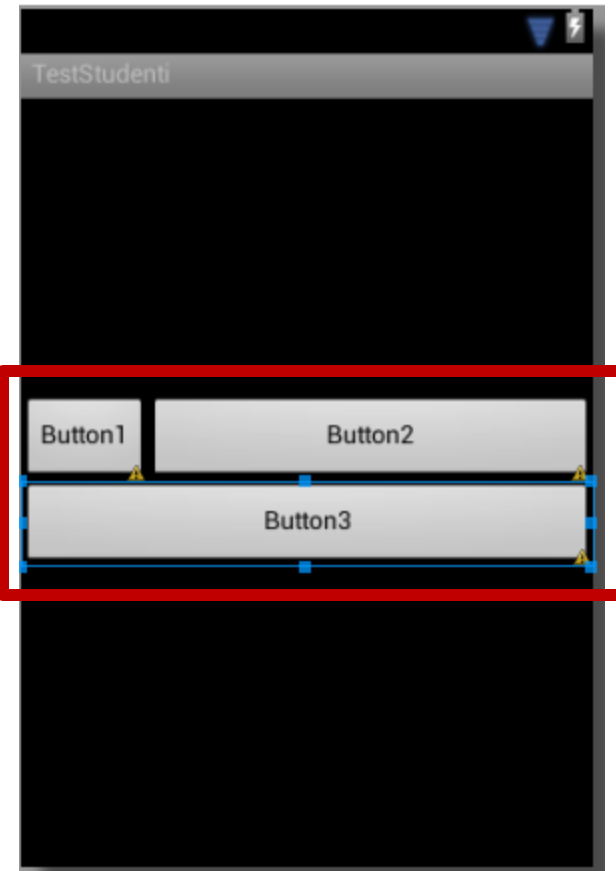
Linear Layout

```xml
<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="bottom"
    android:orientation="vertical" >
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal" >
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button1" />
        <Button
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Button2" />
    </LinearLayout>
    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Button3" />
</LinearLayout>
```

Linear Layout

```xml
<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="center"
    android:orientation="vertical" >
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal" >
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button1" />
        <Button
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Button2" />
    </LinearLayout>
    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Button3" />
</LinearLayout>
```

Linear Layout

```xml
<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="center"
    android:orientation="vertical" >
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal" >
        <Button
            android:layout_width="150px"
            android:layout_height="wrap_content"
            android:text="Button1" />
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button2" />
    </LinearLayout>
    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Button3" />
</LinearLayout>
```
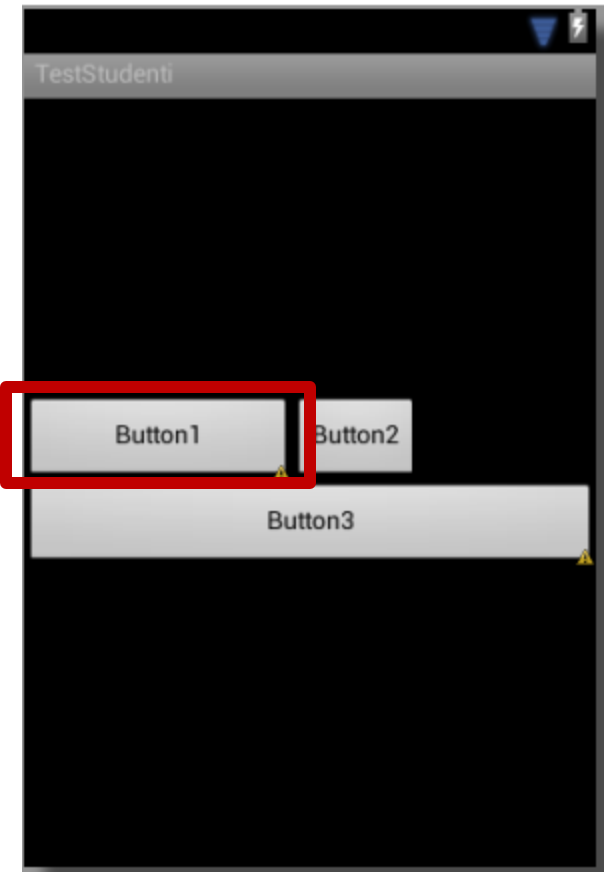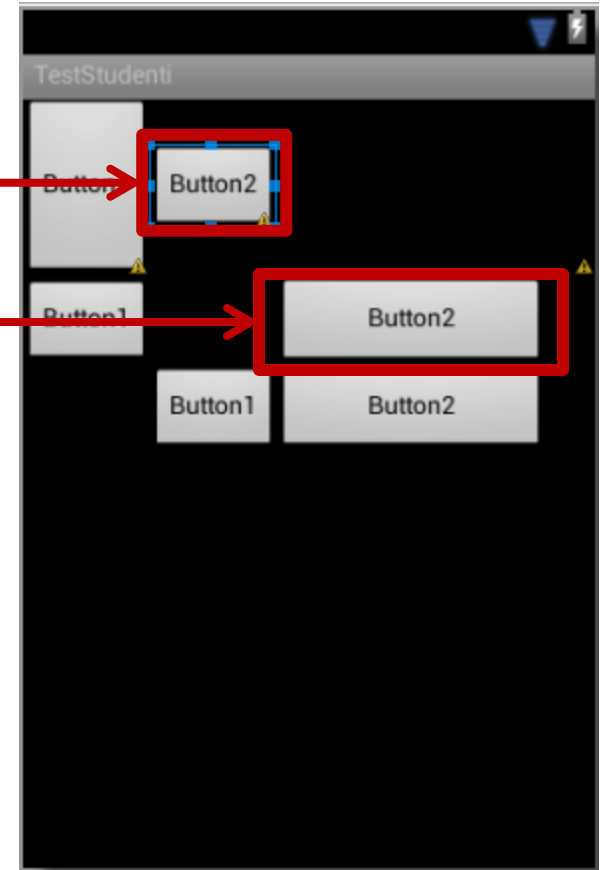
Table Layout

```
<TableLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
    <TableRow>
      <Button android:layout_width="wrap_content"
              android:layout_height="100px"
              android:text="Button1" />
      <Button android:layout_width="10px"
              android:text="Button2" />
    </TableRow>
    <TableRow>
      <Button android:text="Button1" />
      <Button android:layout_width="150px"
              android:layout_height="50px"
              android:layout_column="2"
              android:text="Button2" />
    </TableRow>
    <TableRow>
      <Button android:layout_column="1"
              android:text="Button1" />
      <Button android:text="Button2" />
    </TableRow>
</TableLayout>
```


Column size is set as the biggest width from all of the elements/View from that column

Table Layout

```xml
<TableLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
    <TableRow>
        <Button android:layout_width="wrap_content"
                android:layout_height="100px"
                android:text="Button1" />
        <Button android:layout_width="10px"
                android:text="Button2" />
    </TableRow>
    <TableRow android:layout_gravity="right" >
        <Button android:text="Button1" />
        <Button android:layout_width="150px"
                android:layout_height="50px"
                android:layout_column="2"
                android:text="Button2" />
    </TableRow>
    <TableRow>
        <Button android:layout_column="1"
                android:text="Button1" />
        <Button android:text="Button2" />
    </TableRow>
</TableLayout>
```
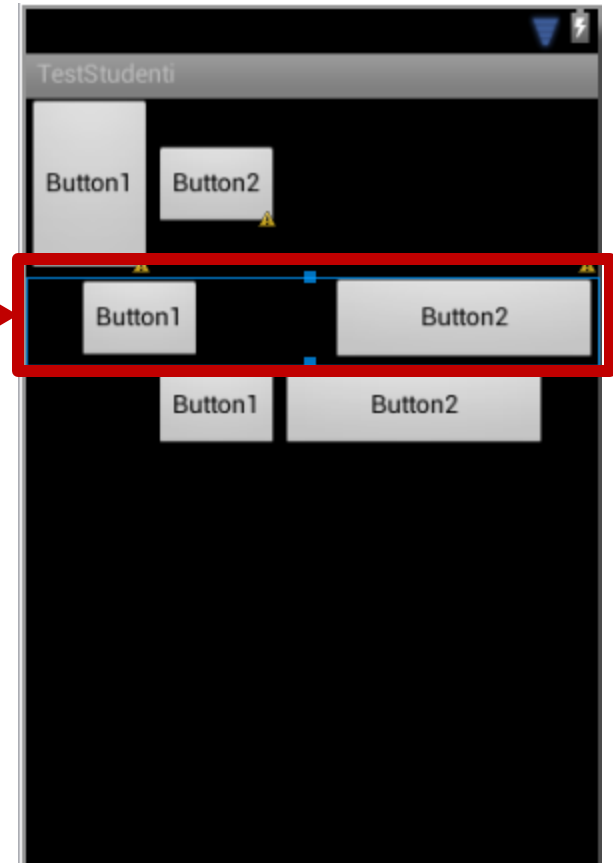
Table Layout

```xml
<TableLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:shrinkColumns="*"
    android:stretchColumns="*" >

    <TableRow android:layout_width="fill_parent"
            android:gravity="center_horizontal">
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button1" />
    </TableRow>

    <TableRow android:gravity="right" >
        <Button android:text="Button1" />
        <Button android:text="Button2" />
    </TableRow>

</TableLayout>
```

Table Layout

```xml
<TableLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <TableRow android:layout_width="fill_parent">
        <Button
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_span="2"
            android:text="Button1" />
        <Button android:text="Button2" />
    </TableRow>

    <TableRow android:gravity="right" >
        <Button android:text="Button3" />
        <Button android:text="Button4" />
        <Button android:text="Button5" />
    </TableRow>

</TableLayout>
```
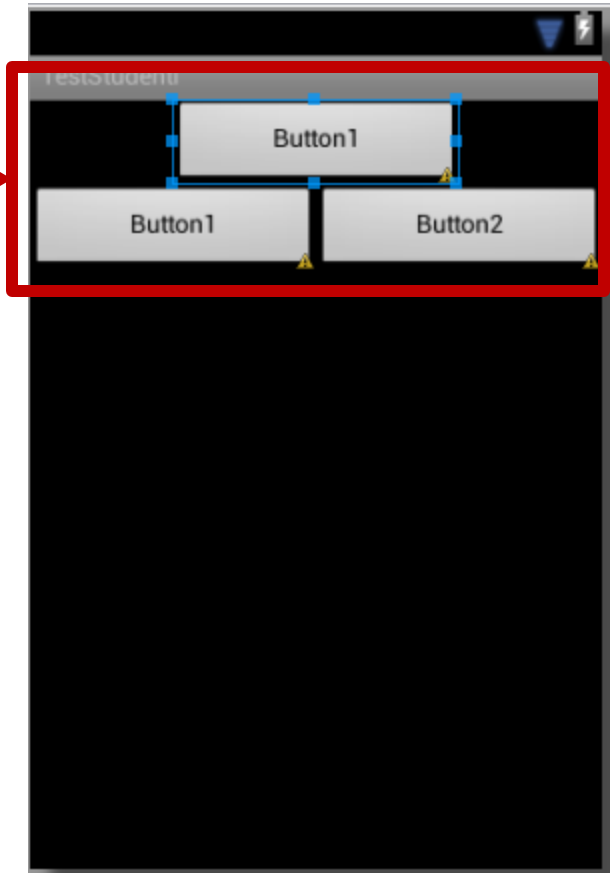
Table Layout

```xml
<TableLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:shrinkColumns="*"
    android:stretchColumns="*"
>

    <TableRow android:layout_width="fill_parent">
        <Button
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_span="2"
            android:text="Button1" />
        <Button android:text="Button2" />
    </TableRow>

    <TableRow android:gravity="right" >
        <Button android:text="Button3" />
        <Button android:text="Button4" />
        <Button android:text="Button5" />
    </TableRow>
</TableLayout>
```
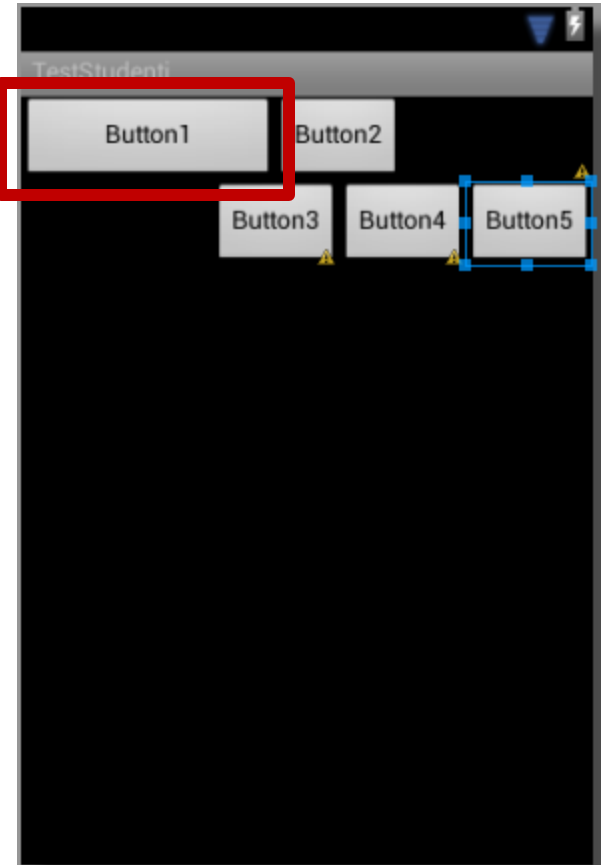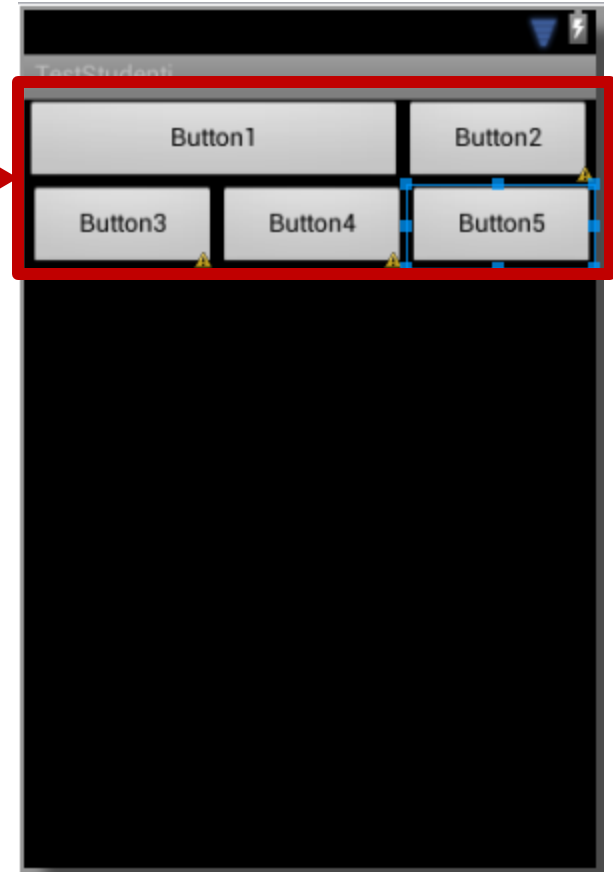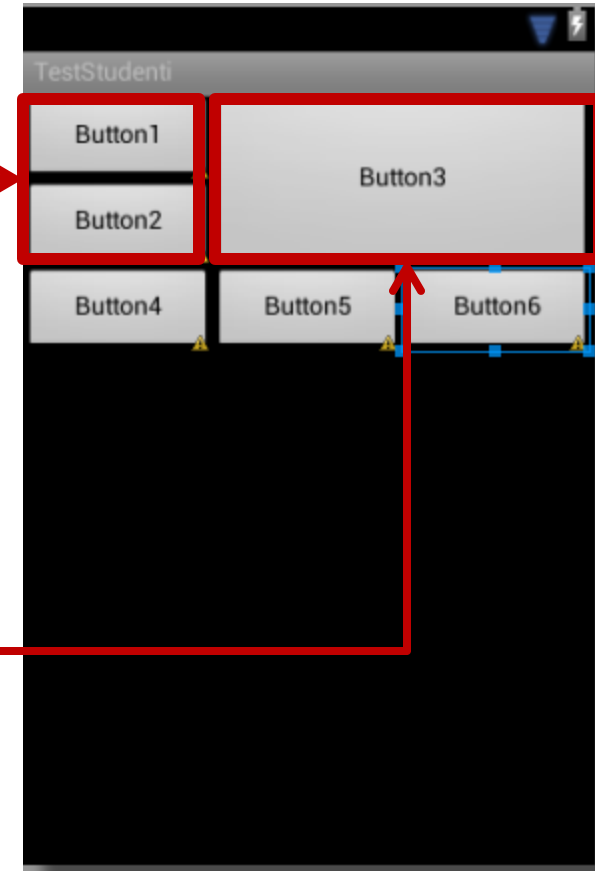
Table Layout

```xml
<TableLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:stretchColumns="*"  >
    <TableRow android:layout_width="fill_parent">
        <LinearLayout android:orientation="vertical"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content">
            <Button android:text="Button1"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content" />
            <Button android:text="Button2"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content" />
        </LinearLayout>
        <Button android:text="Button3" android:layout_span="2"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"/>
    </TableRow>
    <TableRow >
        <Button android:text="Button4" />
        <Button android:text="Button5" />
        <Button android:text="Button6" />
    </TableRow>
</TableLayout>
```

Relative Layout

```xml
<RelativeLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginLeft="28dp"
        android:layout_marginTop="32dp"
        android:text="Button1" />
    <Button
        android:id="@+id/button4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBottom="@+id/button2"
        android:layout_alignTop="@+id/button1"
        android:layout_marginLeft="29dp"
        android:layout_toRightOf="@+id/button2"
        android:text="Button4" />
     <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/button1"
        android:layout_centerHorizontal="true"
```
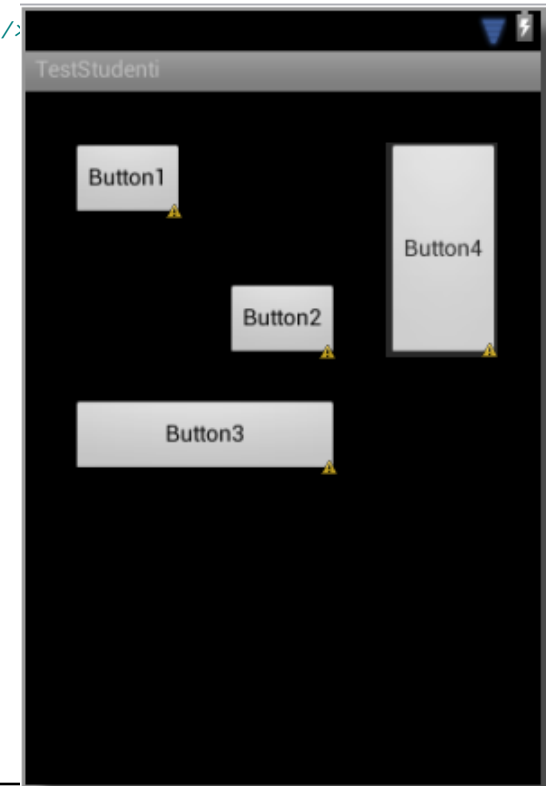
```xml
        android:layout_marginTop="39dp"
        android:text="Button2" />
    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/button1"
        android:layout_centerVertical="true"
        android:layout_toLeftOf="@+id/button4"
        android:text="Button3" />
</RelativeLayout>
```

Relative Layout

- **XML Attributes**
  - layout_toRightOf
  - layout_toLeftOf
  - layout_above
  - layout_below
  - layout_alignLeft
  - layout_alignTop
  - layout_alignRight
  - layout_alignBottom
  - layout_alignParentLeft
  - layout_alignParentRight
  - layout_alignParentTop
  - layout_alignParentBottom
  - layout_marginLeft
  - layout_marginRight
  - layout_marginTop
  - layout_marginBottom