

Lab handout 03

Mobile Application Development (CSE-426), Fall 2022, BE-EE(CS)-VIII

Name: Shakir Ali	CMS ID#: 031-19-0009	Instructor: Dr. Abdul Aziz
Section: A Lab group: NA	Department: Electrical Engineering	Marks obtained out of 100%

NOTE: Must follow submission instructions

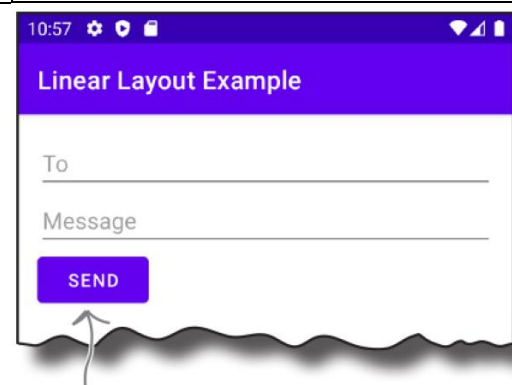
Understanding and using linear layout

Lab Exercise and Submission

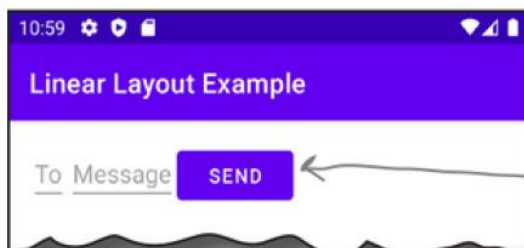
Exercise 1

Design and demonstrate linear layout using following attributes to produce similar kind of results as shown in the figures below.

android:layout_width= android:layout_height=	"wrap_content" "match_parent" "16dp"
android:orientation=	"vertical" "horizontal"



If the orientation is vertical, views are arranged in a single vertical column.



If the orientation is horizontal, views are arranged in a horizontal row. Here, the device language is set to English, so the views are arranged from left to right.

Attach AVD application output here:



Insert XML file code here:

```
<?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:padding="16dp"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/editTextTo"
        android:layout_width="50dp"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="@string/toMail"
        android:inputType="textPersonName" />

    <EditText
        android:id="@+id/editTextMessage"
        android:layout_width="85dp"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="@string/messageText"
        android:inputType="text" />

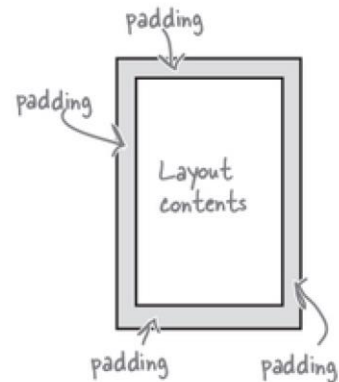
    <Button
        android:id="@+id/btnSend"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/btnSend" />

</LinearLayout>
```

Design and demonstrate linear layout using following attributes to produce similar kind of results as shown in the figures below.

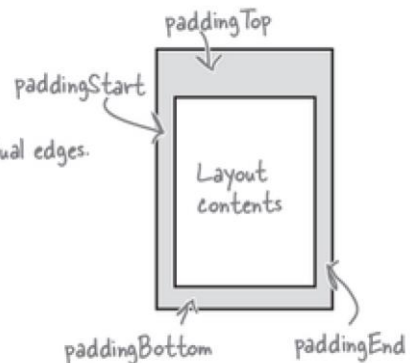
<code>android:padding=</code> <code>android:paddingTop</code> <code>android:paddingStart=</code> <code>android:paddingEnd=</code> <code>android:paddingBottom=</code> <code>android:paddingHorizontal=</code> <code>android:paddingVertical=</code>	<code>"16dp"</code> <code>"32dp"</code>
---	--

```
<LinearLayout ...
    android:padding="16dp" > ← This adds the same padding
                              to all edges of the layout.
</LinearLayout>
```

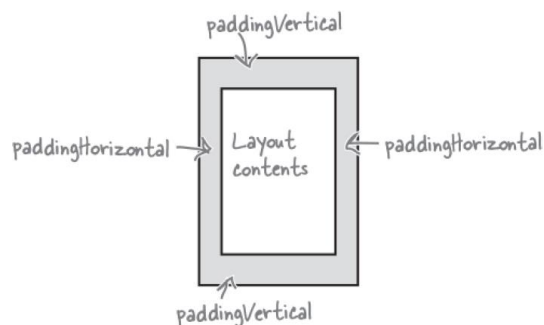


```
<LinearLayout ...
    android:paddingBottom="16dp"
    android:paddingStart="16dp"
    android:paddingEnd="16dp"
    android:paddingTop="32dp" >
</LinearLayout>
```

← Add padding to the individual edges.



attribute adds



Screenshot of AVD output here:



Insert XML codes here:

```
<?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:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="32dp"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Lab #03: Task 02"
        android:textSize="20dp"
    />

</LinearLayout>
```

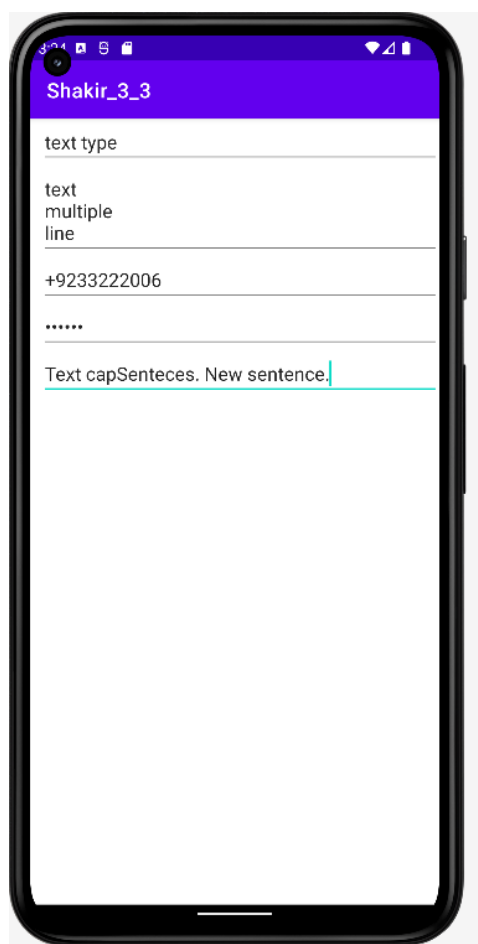
Exercise 3

With linear layout use multiple EditText elements in a single application to demonstrate the effect of following EditText attributes as shown in the table below. Use same hint as the inputType.

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="text"
    android:inputType="text" />
```

Value	What it does
text	Lets the user enter a single line of text.
textMultiLine	Lets the user enter multiple lines of text.
phone	Provides a phone number keypad.
textPassword	Displays a text entry keypad, and your input is concealed.
<p>You can find out more in the online Android developer documentation at https://developer.android.com/training/keyboard-input/style.</p>	
textCapSentences	Capitalizes the first word of a sentence.

Screenshot of AVD output here:



Insert XML codes here:

```
<?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"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/editTypeText"
        android:layout_width="match_parent"
```

```

        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="text"
        android:inputType="text" />

<EditText
    android:id="@+id/editTypeTextMultiLine"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="textMultiLine"
    android:inputType="textMultiLine" />

<EditText
    android:id="@+id/editTypePhone"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="phone"
    android:inputType="phone" />

<EditText
    android:id="@+id/editTypeTextPassword"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="textPassword"
    android:inputType="textPassword" />

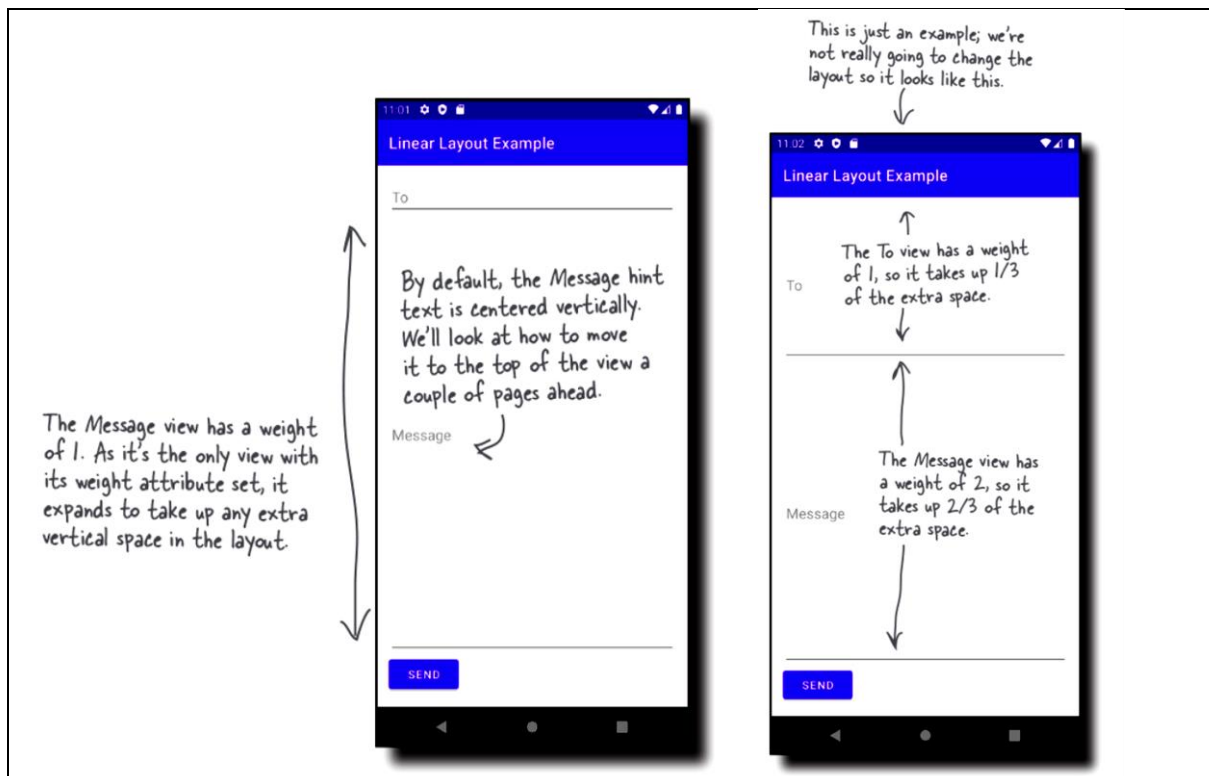
<EditText
    android:id="@+id/editTypeTextCapSentences"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="textCapSentences"
    android:inputType="textCapSentences" />

</LinearLayout>

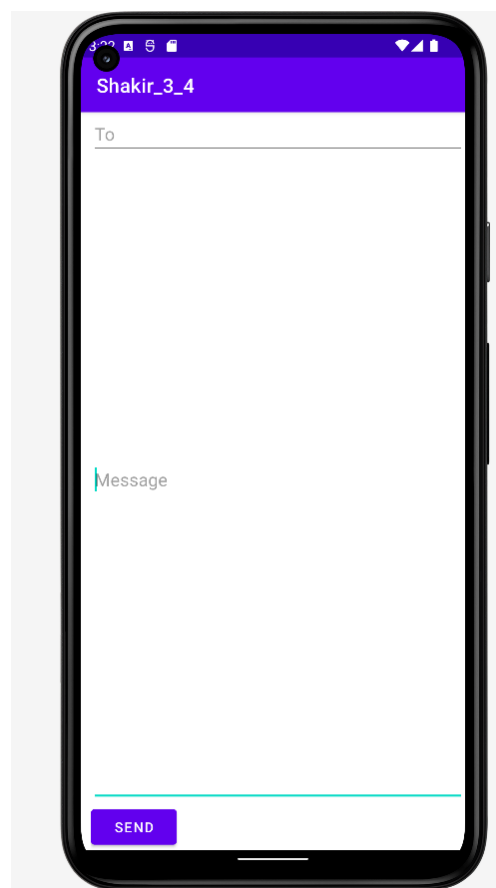
```

Exercise 4

Design a linear layout UI to demonstrate the weight attribute for middle element EditText as shown in the figure below.



Screenshot of AVD output here:



Insert XML codes here:

```
<?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:id="@+id/linearLayout"
    android:layout_width="match parent"
```

```

        android:layout_height="match_parent"
        android:orientation="vertical"
        android:paddingLeft="10dp"
        tools:context=".MainActivity">

        <EditText
            android:id="@+id/editTextTo"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:hint="To"
            android:inputType="textPersonName" />

        <EditText
            android:id="@+id/editTextMessage"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:layout_weight="1"
            android:hint="Message"
            android:inputType="text" />

        <Button
            android:id="@+id/btnSend"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Send"
            />

    </LinearLayout>

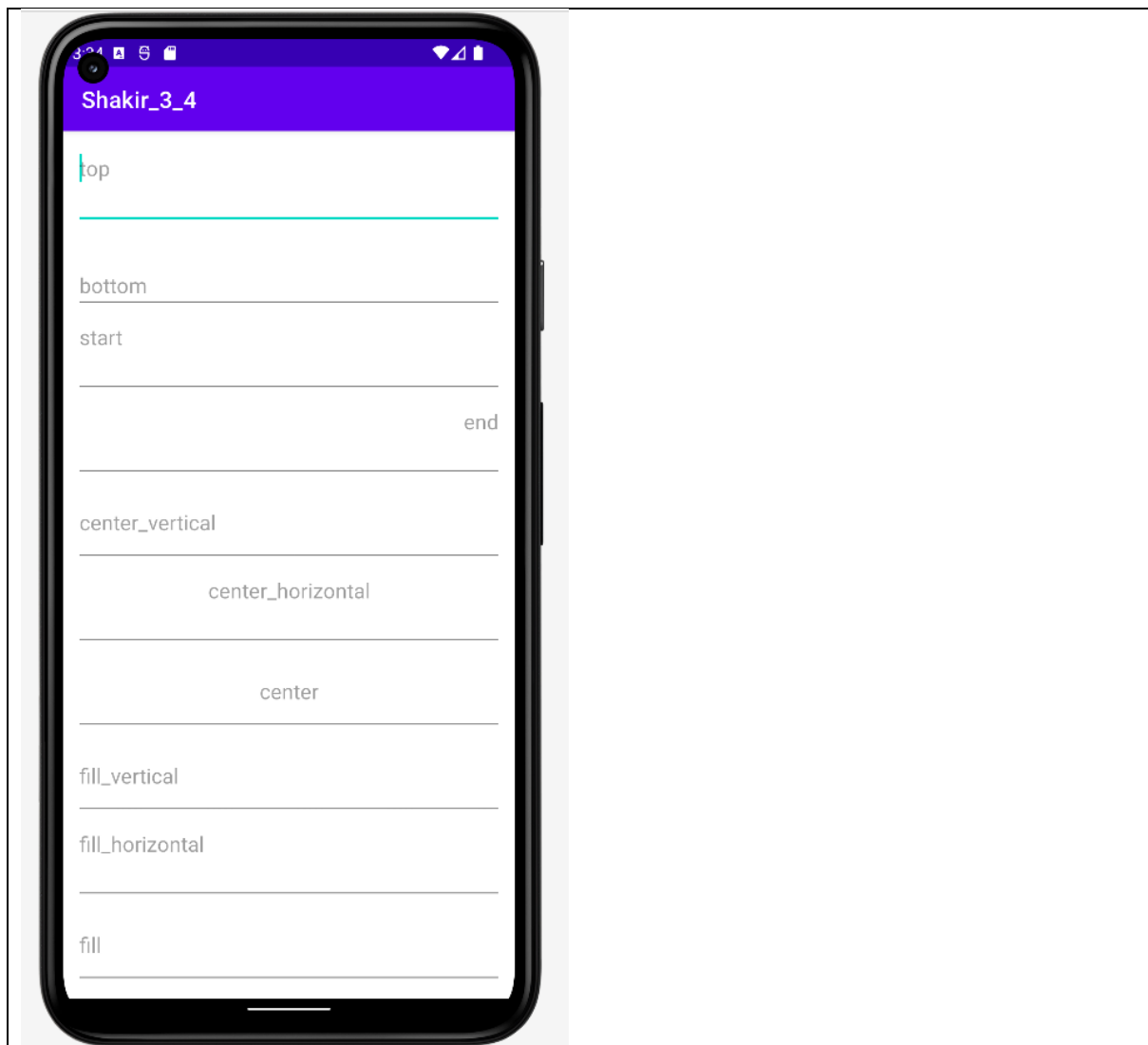
```

Exercise 5

Design a linear layout UI to demonstrate the gravity attribute with following values for EditText element.

top	Puts the view's contents at the top of the view.
bottom	Puts the view's contents at the bottom of the view.
start	Puts the view's contents at the start of the view.
end	Puts the view's contents at the end of the view.
center_vertical	Centers the view's contents vertically.
center_horizontal	Centers the view's contents horizontally.
center	Centers the view's contents vertically and horizontally.
fill_vertical	Makes the view's contents fill the view vertically.
fill_horizontal	Makes the view's contents fill the view horizontally.
fill	Makes the view's contents fill the view vertically and horizontally.

Screenshot of AVD output here:



Insert XML codes here:

```
<?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:id="@+id/linearLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="10dp"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/editTextGravityTop"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:gravity="top"
        android:layout_weight="1"
        android:hint="top"
        android:inputType="text" />

    <EditText
        android:id="@+id/editTextGravityBottom"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:gravity="bottom"
        android:layout_weight="1"
```

```

        android:hint="bottom"
        android:inputType="text" />
<EditText
    android:id="@+id/editTextGravityStart"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:gravity="start"
    android:layout_weight="1"
    android:hint="start"
    android:inputType="text" />

<EditText
    android:id="@+id/editTextGravityEnd"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:gravity="end"
    android:layout_weight="1"
    android:hint="end"
    android:inputType="text" />

<EditText
    android:id="@+id/editTextGravityCenterVertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:gravity="center_vertical"
    android:layout_weight="1"
    android:hint="center_vertical"
    android:inputType="text" />

<EditText
    android:id="@+id/editTextGravityCenterHorizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:gravity="center_horizontal"
    android:layout_weight="1"
    android:hint="center_horizontal"
    android:inputType="text" />

<EditText
    android:id="@+id/editTextGravityCenter"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:gravity="center"
    android:layout_weight="1"
    android:hint="center"
    android:inputType="text" />

<EditText
    android:id="@+id/editTextGravityFillVertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:gravity="fill_vertical"
    android:layout_weight="1"
    android:hint="fill_vertical"
    android:inputType="text" />

<EditText
    android:id="@+id/editTextGravityFillHorizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"

```

```
        android:ems="10"
        android:gravity="fill_horizontal"
        android:layout_weight="1"
        android:hint="fill_horizontal"
        android:inputType="text" />
```

```
<EditText
    android:id="@+id/editTextGravityFill"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:gravity="fill"
    android:layout_weight="1"
    android:hint="fill"
    android:inputType="text" />
```

```
</LinearLayout>
```