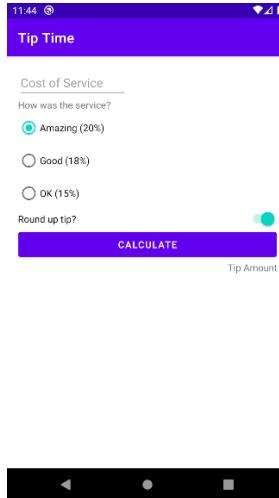


## Practical class # 6 – Tip Calculator – Part 1 – XML Layout

In this practical class we will discuss how to create an app that interacts with user. The app should look like this.



### 1) Create an empty activity and identify the Views that we will use

We will use these Views:

- EditText - for entering and editing text
- TextView - to display text like the service question and tip amount
- RadioButton - a selectable radio button for each tip option
- RadioGroup - to group the radio button options
- Switch - an on/off toggle for choosing whether to round up the tip or not

Name the app “Tip Time”.

### 2) Try to understand XML

Open res > layout > activity\_main.xml. On the top right corner, select code instead of View.

### 3) Create the layout using XML

On the top right corner, select Split, to see both the code and the layout. Check that selecting Views in the Design editor highlights the code related to the selected View.

### 4) Add cost of service text field and the other Views

Check the EditText documentation at the following URL:

<https://developer.android.com/reference/kotlin/android/widget/EditText>

Find the part related to XML and paste it in android studio and after that delete the TextView.

This View will now be highlighted in red since it is not constrained.

Add a padding to the ViewGroup.

android:padding="16dp"

Add these two constraints to anchor the EditText.

```
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"
```

Notice that layout\_height and layout\_width are set to "wrap\_content". Fix layout\_width to 160dp.  
Notice that inputType is set to "text", change it to "numberDecimal". Add an hint attribute with value "Cost of Service".

android:hint="Cost of Service"

Now insert a TextView with this code and rename the EditText as cost\_of\_service

```
<TextView  
    android:id="@+id/service_question"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="How was the service?"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@id/cost_of_service"/>
```

Add the RadioGroup.

```
<RadioGroup  
    android:id="@+id/tip_options"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:orientation="vertical"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@id/service_question">  
  
    <!-- add RadioButtons here -->  
  
</RadioGroup>
```

Then Insert the three Radio buttons.

```
<RadioButton  
    android:id="@+id/option_twenty_percent"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Amazing (20%)" />  
  
<RadioButton  
    android:id="@+id/option_eighteen_percent"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Good (18%)" />  
  
<RadioButton  
    android:id="@+id/option_fifteen_percent"
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Okay (15%)" />
```

Run the app and check if it works.

Add a default selection to the RadioGroup

```
    android:checkedButton="@+id/option_twenty_percent"
```

Add a Switch to round up the tip.

```
<Switch
    android:id="@+id/round_up_switch"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:checked="true"
    android:text="Round up tip?"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="@+id/tip_options"
    app:layout_constraintTop_toBottomOf="@+id/tip_options" />
```

Finally, add calculate button and tip result Views.

```
<Button
    android:id="@+id/calculate_button"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="Calculate"
    app:layout_constraintTop_toBottomOf="@+id/round_up_switch"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />
```

```
<TextView
    android:id="@+id/tip_result"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/calculate_button"
    android:text="Tip Amount" />
```

## 5) Extract the hardcoded Strings

Over on hardcoded Strings in the XML to show the hint on resolving this issue.

## 6) Reformat the XML code

Choose code > Reformat code to automatically reformat the code.

