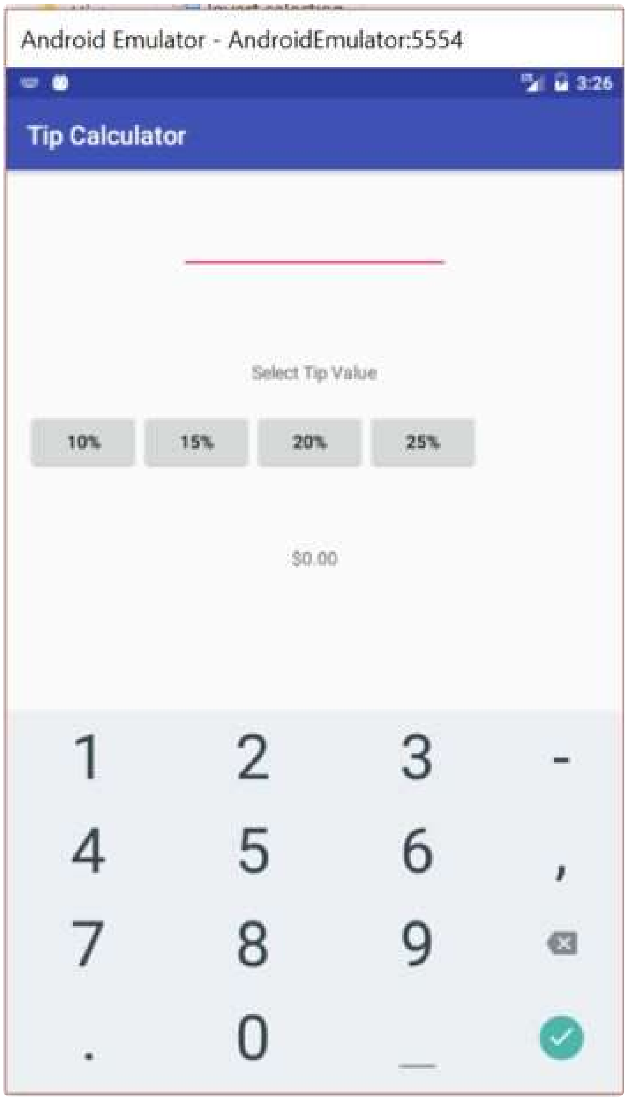
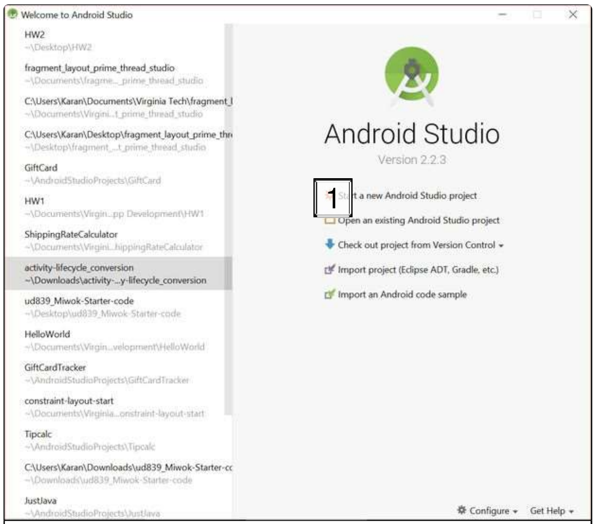
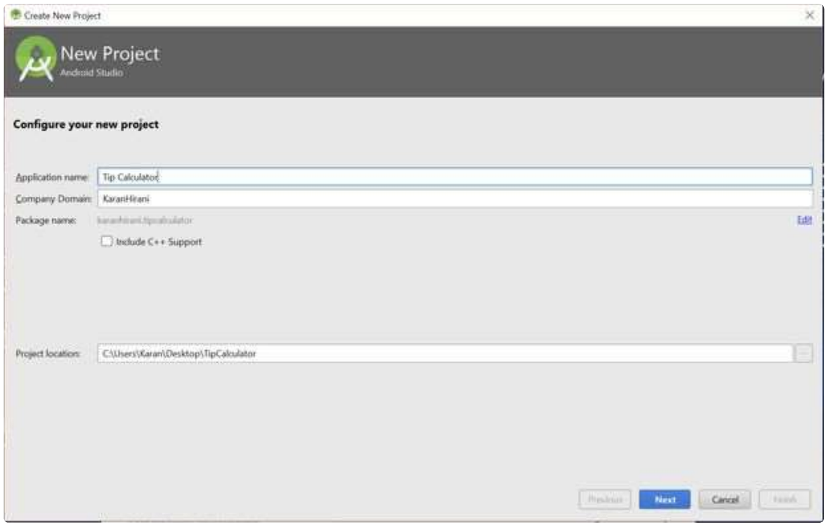


Step 1: Creating the Project

Start a new Android Studio Project. Use the pictures and notes as reference to guide you. Make sure to follow same naming procedure to avoid confusion in later steps. Once you are all finished, you should see a screen like the last image.


Welcome to Android Studio
HW2
~\Desktop\HW2
fragment_layout_prime_thread_studio
~\Documents\fragme..._prime_thread_studio
C:\Users\Karan\Documents\Virginia Tech\fragment_l
~\Documents\Virgini...t_prime_thread_studio
C:\Users\Karan\Desktop\fragment_layout_prime_thr
~\Desktop\fragment_...t_prime_thread_studio
GiftCard
~\AndroidStudioProjects\GiftCard
HW1
~\Documents\Virgin...pp Development\HW1
ShippingRateCalculator
~\Documents\Virgini...hippingRateCalculator
activity-lifecycle_conversion
~\Downloads\activity-...y-lifecycle_conversion
ud839_Miwok-Starter-code
~\Desktop\ud839_Miwok-Starter-code
HelloWorld
~\Documents\Virgin...velopment\HelloWorld
GiftCardTracker
~\AndroidStudioProjects\GiftCardTracker
constraint-layout-start
~\Documents\Virginia...onstraint-layout-start
Tipcalc
~\AndroidStudioProjects\Tipcalc
C:\Users\Karan\Downloads\ud839_Miwok-Starter-c
~\Downloads\ud839_Miwok-Starter-code
JustJava
~\AndroidStudioProjects\JustJava
Android Studio
Version 2.2.3
1
Start a new Android Studio project
Open an existing Android Studio project
Check out project from Version Control
Import project (Eclipse ADT, Gradle, etc.)
Import an Android code sample
Configure
Get Help

1. Select Start a New Android Studio Project



Configure your new project

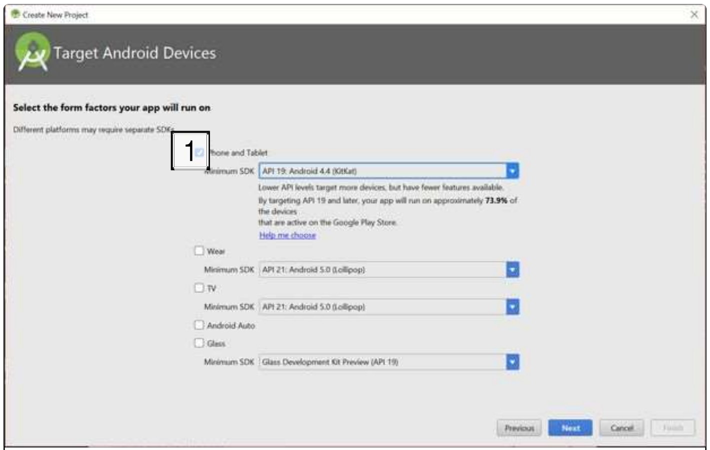
Application name: Tip Calculator

Company Domain: KaranHirani

Package name: karanhirani.tipcalculator

☐ Include C++ Support

Project location: C:\Users\Karan\Desktop\TipCalculator



Select the form factors your app will run on

Different platforms may require separate SDKs

☑ Phone and Tablet

Minimum SDK: API 19: Android 4.4 (KitKat)

Lower API levels target more devices, but have fewer features available. By targeting API 19 and later, your app will run on approximately **73.9%** of the devices that are active on the Google Play Store.

Help me choose

☐ Wear

Minimum SDK: API 21: Android 5.0 (Lollipop)

☐ TV

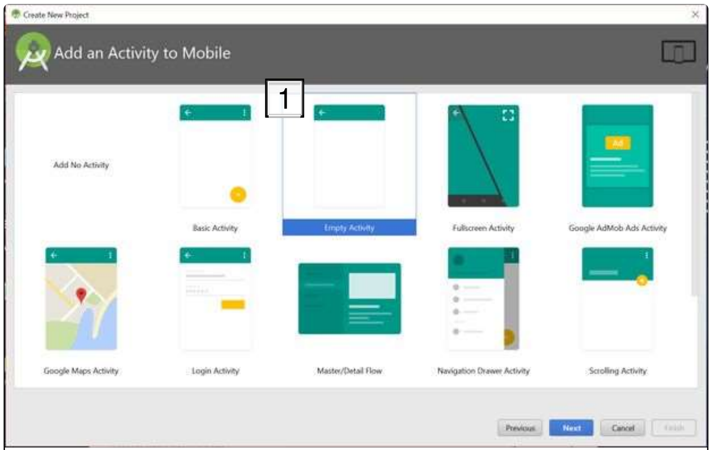
Minimum SDK: API 21: Android 5.0 (Lollipop)

☐ Android Auto

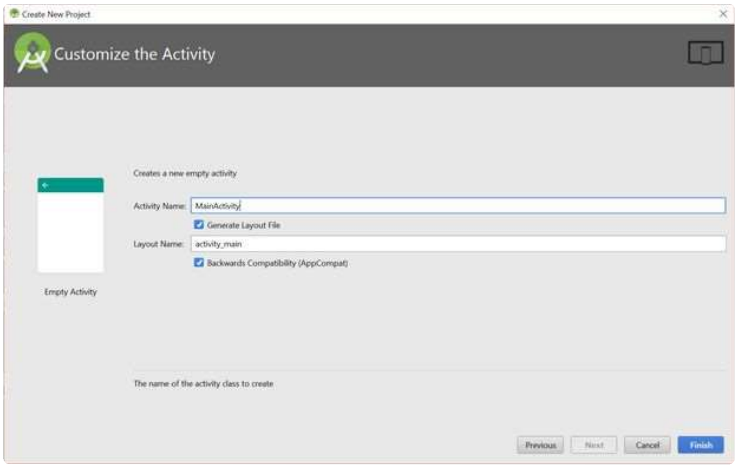
☐ Glass

Minimum SDK: Glass Development Kit Preview (API 19)

1. Make sure to have minimum of 4.4 SDK



1. Select Empty Activty



Creates a new empty activity

Activity Name: MainActivity

☑ Generate Layout File

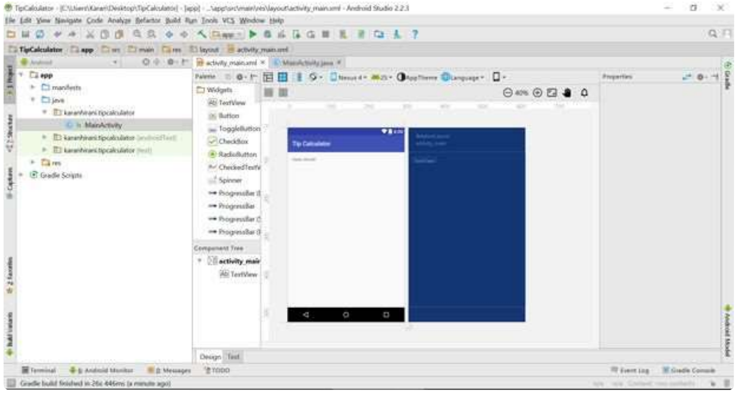
Layout Name: activity_main

☑ Backwards Compatibility (AppCompat)

The name of the activity class to create

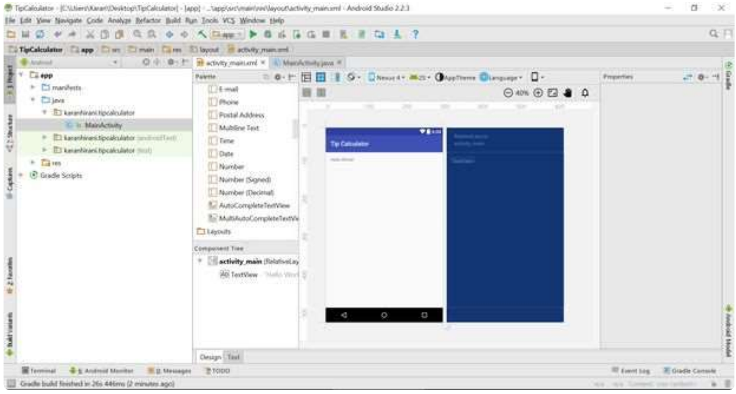
Step 2: Go to the XML File

Click on the activity_main.xml tab at the top of the screen.



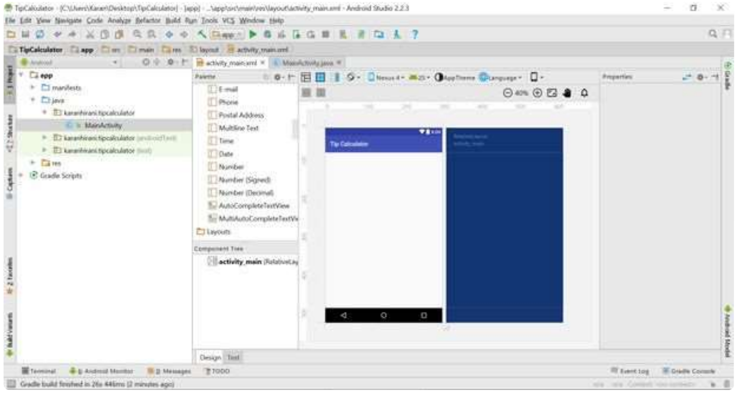
Step 3: Familiarize Yourself With the XML

XML is the front end part of the project. It creates everything that you see on the screen, while the Java part makes the XML "do things". Android studio has created a drag and drop system that lets you add different types of things on the screen, like spinners, images, buttons, etc. Scroll down till you find Number (Decimal).



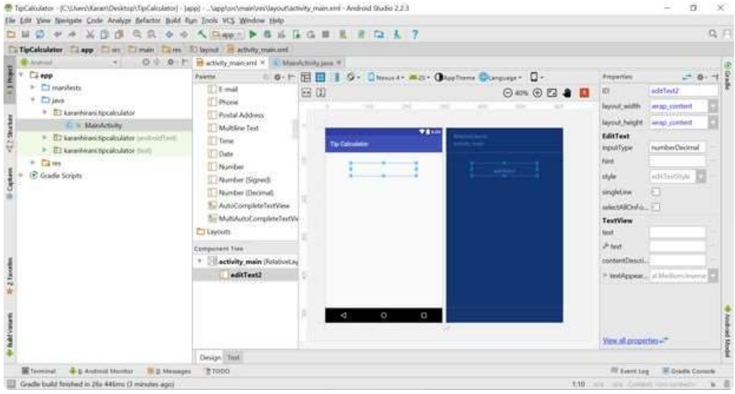
Step 4: Delete Hello World Text

Delete the hello world text on the screen by clicking on it and then pressing delete.


TipCalculator - [C:\Users\Karan\Desktop\TipCalculator] - [app] - ...\app\src\main\res\layout\activity_main.xml - Android Studio 2.2.3
Palette
E-mail
Phone
Postal Address
Multiline Text
Time
Date
Number
Number (Signed)
Number (Decimal)
AutoCompleteTextView
MultiAutoCompleteTextView
Layouts
Component Tree
activity_main (RelativeLayout)
Tip Calculator
Properties
Design
Text

Step 5: Drag and Drop Number (decimal) Text Field

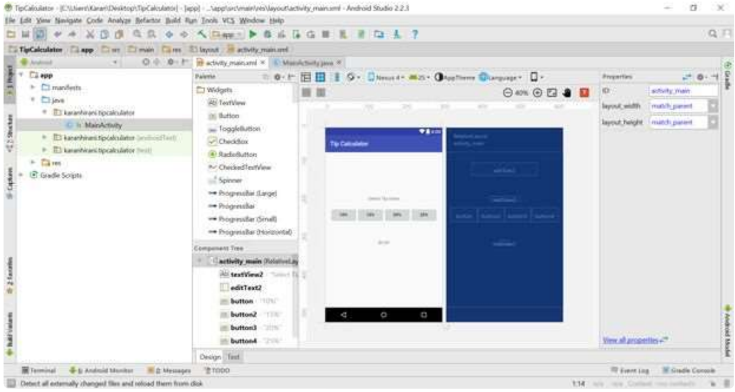
Drag the Number(Decimal) widget on to the screen. Drag it somewhere near the top of the screen. This will allow the user to input decimal numbers. Refer to the picture in the intro for reference.



Properties
ID: editText2
layout_width: wrap_content
layout_height: wrap_content
EditText
inputType: numberDecimal
hint:
style: editTextStyle
singleLine: □
selectAllOnFo...: □
TextView
text:
text:
contentDescri...:
textAppear...: @:Medium.Inverse

Step 6: Add the Other Elements to the Screen

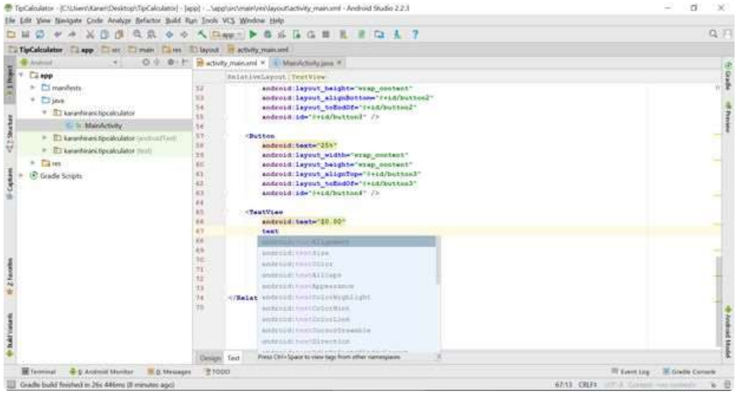
Drag and drop the different buttons and textfields on to the screen so it looks like something in the picture. If you have any questions of which type of widget to use, look at the blue screen on the right to see the widget I used. If you double click on any of the widgets, you will be able to change the text of it. Change the text to the appropriate numbers and words.



Properties
ID: activity_main
layout_width: match_parent
layout_height: match_parent

Step 7: Customize the XML Text File

If you are familiar with XML code, you can go ahead and customize the widgets on the screen to your liking. If you are unfamiliar with XML code, start typing something like "textSize" and android studio will pull up suggestions of the properties that you would like to change.

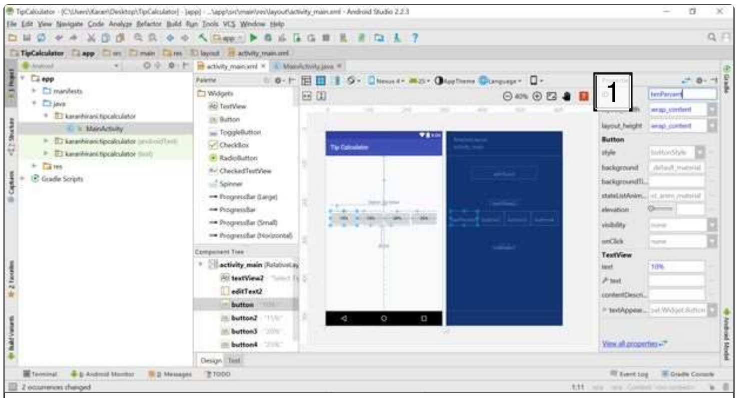


```
            android:layout_height="wrap_content"
            android:layout_alignBottom="@+id/button2"
            android:layout_toEndOf="@+id/button2"
            android:id="@+id/button3" />

        <Button
            android:text="25%"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignTop="@+id/button3"
            android:layout_toEndOf="@+id/button3"
            android:id="@+id/button4" />

        <TextView
            android:text="$0.00"
            text
                android:textAlignment
                android:textSize
                android:textColor
                android:textAllCaps
                android:textAppearance
                android:textColorHighlight
                android:textColorHint
                android:textColorLink
                android:textCursorDrawable
                android:textDirection

    </Relat
```

Press Ctrl+Space to view tags from other namespaces

Step 8: Change IDs of Buttons

In the top right corner, you should see a Properties section. In the properties section, the first field is title ID. For each of the buttons, name the ID so that it corresponds with the number. For example, change the ID of the 10% button to tenPercent. Make sure not to use numbers for the names.



ID: tenPercent
layout_width: wrap_content
layout_height: wrap_content
text: 10%

1. Change name for each button

Step 9: Initialize the Buttons and Connect Java With XML

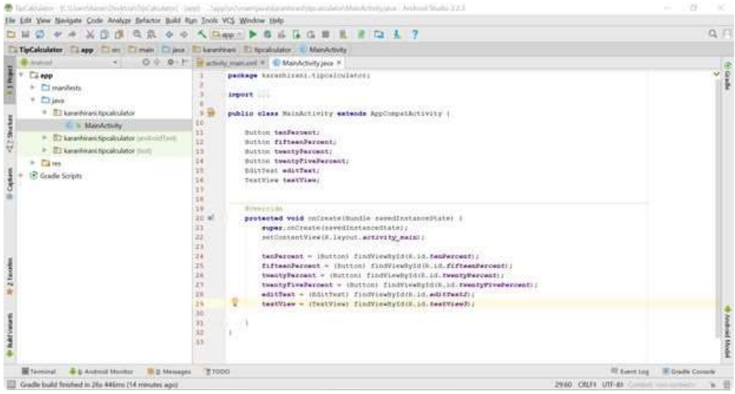
Go ahead and click back on the MainActivity.java tab. Follow the code in the image. What we are doing here is creating variables, or objects, in Java and connecting them with XML so we can control what each XML element does.

What the code below does is it initializes each button as a field in Java. It then connects the Java field with the XML elements.

Code below:

```
Button tenPercent;
 Button fifteenPercent;
Button twentyPercent;
Button twentyFivePercent;
EditText editText;
TextView textView;
Double money;
```

```
 tenPercent = (Button) findViewById(R.id.tenPercent);
 fifteenPercent = (Button) findViewById(R.id.fifteenPercent);
twentyPercent = (Button) findViewById(R.id.twentyPercent);
twentyFivePercent = (Button) findViewById(R.id.twentyFivePercent);
editText = (EditText) findViewById(R.id.editText2);
textView = (TextView) findViewById(R.id.textView3);
```



```
package karanhirani.tipcalculator;

import ...

public class MainActivity extends AppCompatActivity {

    Button tenPercent;
    Button fifteenPercent;
    Button twentyPercent;
    Button twentyFivePercent;
    EditText editText;
    TextView textView;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        tenPercent = (Button) findViewById(R.id.tenPercent);
        fifteenPercent = (Button) findViewById(R.id.fifteenPercent);
        twentyPercent = (Button) findViewById(R.id.twentyPercent);
        twentyFivePercent = (Button) findViewById(R.id.twentyFivePercent);
        editText = (EditText) findViewById(R.id.editText2);
        textView = (TextView) findViewById(R.id.textView3);

    }
}
```

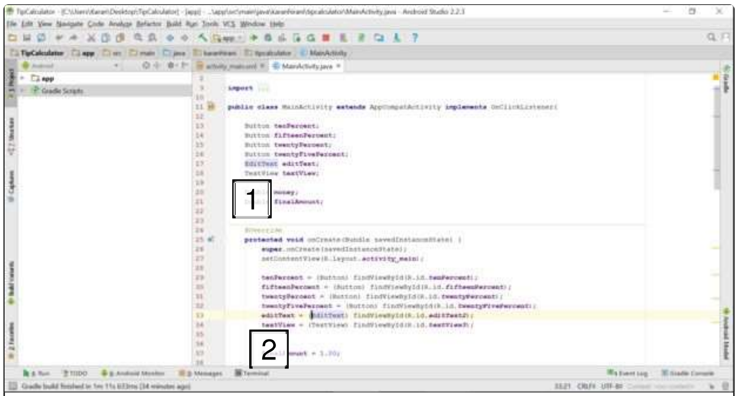
Step 10: Add Two Fields to Keep Track of Your Money and Tip Value

Create two fields to keep track of the input by the user and the final amount to display on the screen.

Code Below:

```
Double money;
 Double finalAmount;
```

```
finalAmount = 1.00;
```



```
import ...

public class MainActivity extends AppCompatActivity implements OnClickListener{

    Button tenPercent;
    Button fifteenPercent;
    Button twentyPercent;
    Button twentyFivePercent;
    EditText editText;
    TextView textView;

    Double money;
    Double finalAmount;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        tenPercent = (Button) findViewById(R.id.tenPercent);
        fifteenPercent = (Button) findViewById(R.id.fifteenPercent);
        twentyPercent = (Button) findViewById(R.id.twentyPercent);
        twentyFivePercent = (Button) findViewById(R.id.twentyFivePercent);
        editText = (EditText) findViewById(R.id.editText2);
        textView = (TextView) findViewById(R.id.textView3);

        finalAmount = 1.00;
```

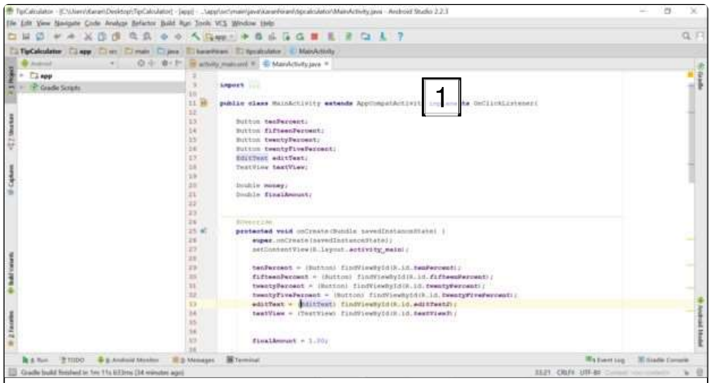
1. Create the follow fields
2. Set finalAmount to the following value.

Step 11: Implement OnClickListnener Interface

Implement the OnClickListener Interface at the top of the screen. This listener will allow us to give the buttons functionality. When you implement the listener, there will be an error. To solve this, press alt + enter on PC or option + enter on Mac. It will ask you to implement the onClick method so just press enter. You should see the method added at the bottom of the screen, shown in the second image.

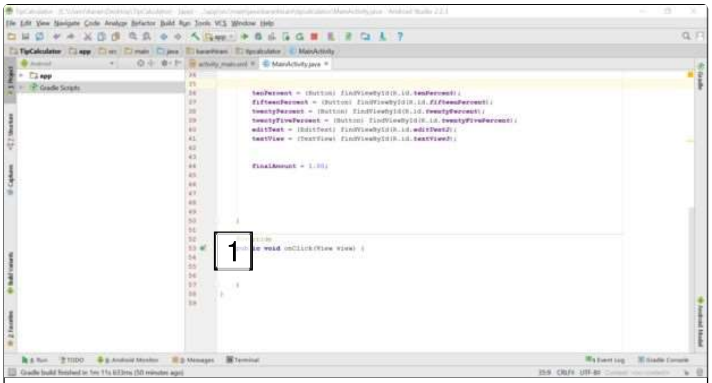
Code below:

```
implements OnClickListener
```



```
import ...

public class MainActivity extends AppCompatActivity implements OnClickListener{

    Button tenPercent;
    Button fifteenPercent;
    Button twentyPercent;
    Button twentyFivePercent;
    EditText editText;
    TextView textView;

    Double money;
    Double finalAmount;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        tenPercent = (Button) findViewById(R.id.tenPercent);
        fifteenPercent = (Button) findViewById(R.id.fifteenPercent);
        twentyPercent = (Button) findViewById(R.id.twentyPercent);
        twentyFivePercent = (Button) findViewById(R.id.twentyFivePercent);
        editText = (EditText) findViewById(R.id.editText2);
        textView = (TextView) findViewById(R.id.textView3);

        finalAmount = 1.00;
```

1. implement the OnClickListener Interface



```
        tenPercent = (Button) findViewById(R.id.tenPercent);
        fifteenPercent = (Button) findViewById(R.id.fifteenPercent);
        twentyPercent = (Button) findViewById(R.id.twentyPercent);
        twentyFivePercent = (Button) findViewById(R.id.twentyFivePercent);
        editText = (EditText) findViewById(R.id.editText2);
        textView = (TextView) findViewById(R.id.textView3);


        finalAmount = 1.00;






    }

    @Override
    public void onClick(View view) {



    }
}
```

1. The following method will automatically be added at bottom of code

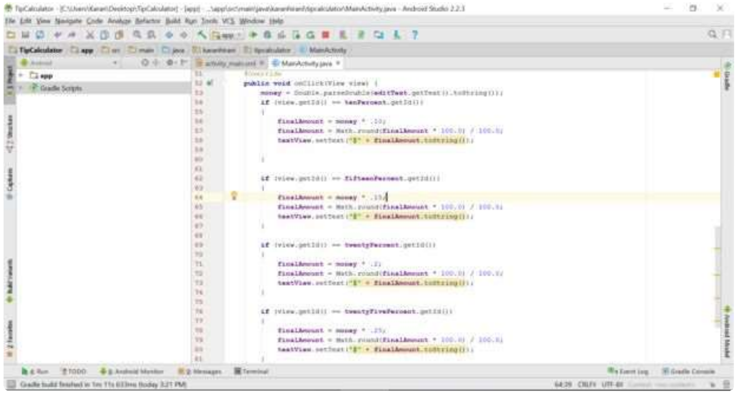
Step 12: Implement OnClick Method

The OnClick Method determines which button was clicked and takes the appropriate action. First, get the monetary value that the user puts into the phone with the following code:

```
money = Double.parseDouble(editText.getText().toString());
```

Second, create an if structure for each of the 4 buttons to determine which one was clicked. Then, within each if structure, set finalAmount equal to money times the percentage of the button clicked. We then have to round the finalAmount to two decimals and set the following value as the text to display. The code below should help you if you need some help implementing the method:

```
if (view.getId() == tenPercent.getId())
 {
    finalAmount = money * .10;
    finalAmount = Math.round(finalAmount * 100.0) / 100.0;
    textView.setText("$" + finalAmount.toString());
}

if (view.getId() == fifteenPercent.getId())
{    finalAmount = money * .15;
     finalAmount = Math.round(finalAmount * 100.0) / 100.0;
     textView.setText("$" + finalAmount.toString());
}

if (view.getId() == twentyPercent.getId())
{    finalAmount = money * .2;
     finalAmount = Math.round(finalAmount * 100.0) / 100.0;
     textView.setText("$" + finalAmount.toString());
}

if (view.getId() == twentyFivePercent.getId())
{
    finalAmount = money * .25;
    finalAmount = Math.round(finalAmount * 100.0) / 100.0;
    textView.setText("$" + finalAmount.toString());
}
```



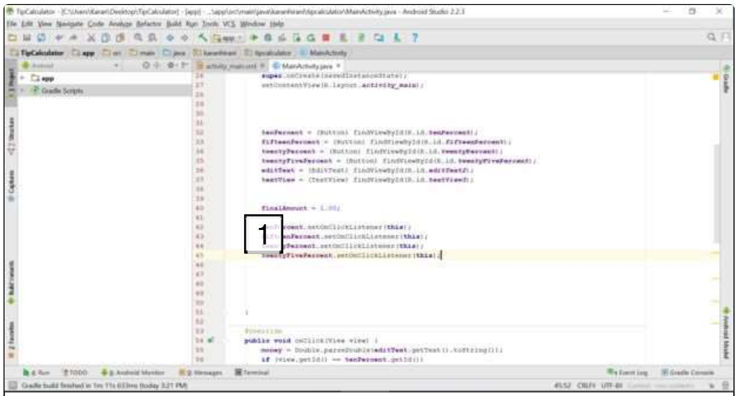
```
@Override
public void onClick(View view) {
    money = Double.parseDouble(editText.getText().toString());
    if (view.getId() == tenPercent.getId())
    {
        finalAmount = money * .10;
        finalAmount = Math.round(finalAmount * 100.0) / 100.0;
        textView.setText("$" + finalAmount.toString());

    }

    if (view.getId() == fifteenPercent.getId())
    {
        finalAmount = money * .15;
        finalAmount = Math.round(finalAmount * 100.0) / 100.0;
        textView.setText("$" + finalAmount.toString());
    }

    if (view.getId() == twentyPercent.getId())
    {
        finalAmount = money * .2;
        finalAmount = Math.round(finalAmount * 100.0) / 100.0;
        textView.setText("$" + finalAmount.toString());
    }

    if (view.getId() == twentyFivePercent.getId())
    {
        finalAmount = money * .25;
        finalAmount = Math.round(finalAmount * 100.0) / 100.0;
        textView.setText("$" + finalAmount.toString());
    }
```

TipCalculator - [C:\Users\Karan\Desktop\TipCalculator] - [app] - ...\app\src\main\java\karanhiram\tipcalculator\MainActivity.java - Android Studio 2.2.3
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
TipCalculator app src main java karanhiram tipcalculator MainActivity
activity_main.xml MainActivity.java
app
Gradle Scripts
TODO
Android Monitor
Messages
Terminal
Event Log
Gradle Console
Gradle build finished in 1m 11s 633ms (today 3:21 PM)
64:29 CRLF UTF-8

Step 13: Make the Buttons Clickable

To make the buttons actually "do things", we have to set their onClickListener method. Since we implemented the interface for OnClickListener in the class, we can pass the argument "this". Add the following code in the onCreate method:

```
tenPercent.setOnClickListener(this);
 fifteenPercent.setOnClickListener(this);
twentyPercent.setOnClickListener(this);
twentyFivePercent.setOnClickListener(this);
```

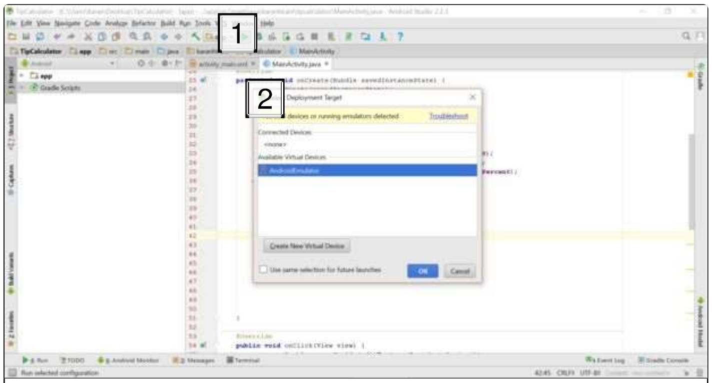


```
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);


        tenPercent = (Button) findViewById(R.id.tenPercent);
        fifteenPercent = (Button) findViewById(R.id.fifteenPercent);
        twentyPercent = (Button) findViewById(R.id.twentyPercent);
        twentyFivePercent = (Button) findViewById(R.id.twentyFivePercent);
        editText = (EditText) findViewById(R.id.editText2);
        textView = (TextView) findViewById(R.id.textView2);


        finalAmount = 1.00;

        tenPercent.setOnClickListener(this);
        fifteenPercent.setOnClickListener(this);
        twentyPercent.setOnClickListener(this);
        twentyFivePercent.setOnClickListener(this);


    }

    @Override
    public void onClick(View view) {
        money = Double.parseDouble(editText.getText().toString());
        if (view.getId() == tenPercent.getId())
```

1. Add the following code

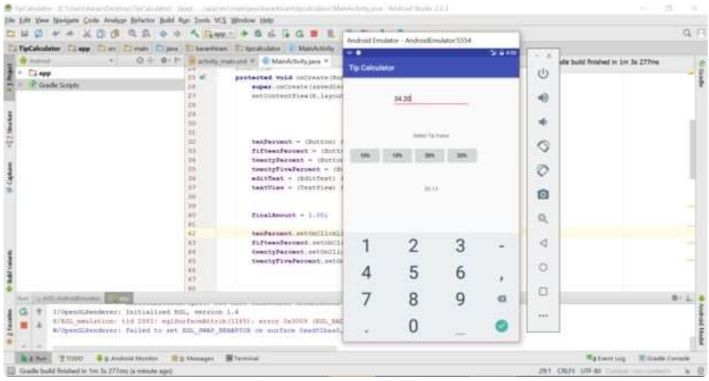
Step 14: Run the Application

We're all done! Now we just have to run the app. If you have an Android device, you can plug it into your computer. To run the app, click the green play button at the top of the screen. Here you can see all your available devices, emulators or android devices. Once you have the app started up, feel free to plug in numbers and test out each of the buttons.



1
2
Deployment Target
No USB devices or running emulators detected
Troubleshoot
Connected Devices
<none>
Available Virtual Devices
AndroidEmulator
Create New Virtual Device
Use same selection for future launches
OK
Cancel

1. Click play button to run app
2. Here, you will be able to see all the devices that are available to run the application.



Step 15: Conclusion

Now that you have created your first application, feel free to customize and add your own features to the application.