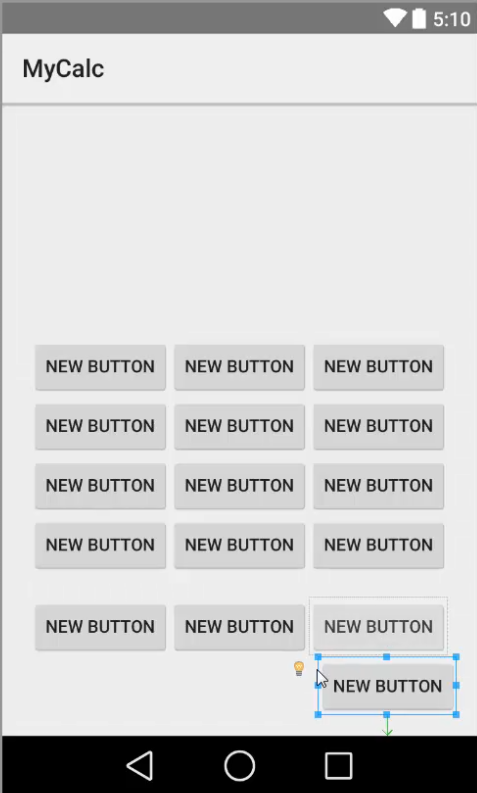
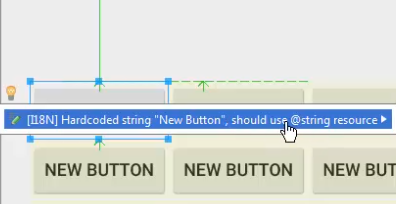
Tutorial 3 – Making a Calculator Application

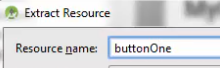
1. Start Android Studio and Create a new project. You may name this however you wish, but would be best if you had “Calculator” somewhere in the title. If you have a project currently open, you will need to go to “file,” and select “Close Project,” which will bring you back to the Android Studio main menu.
2. Delete the Hello World Text Field. Now, make a 3x4 grid of small button widgets from the palette near the middle of the phone design. These will be your buttons 0-9 and C, for clear. Next we will add 4 more buttons underneath our first 3x4 row, but these will be all in the same row. These will be for the add, subtract, multiply, and divide buttons. You may need to put the

last button in the bottom left corner of the screen to fit all the buttons on the screen (we will fix this later). See Below:

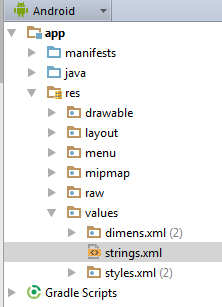
1. Click on the first button, and you will see a yellow light bulb appear just to the left of the button. Click on that light bulb, then click on the text that appears. Android Studio is asking us to set a string name for the strings.xml file. This strings.xml file is useful because it stores all of the projects strings in one file, which one popular usage of this would be to alter the language of your program.



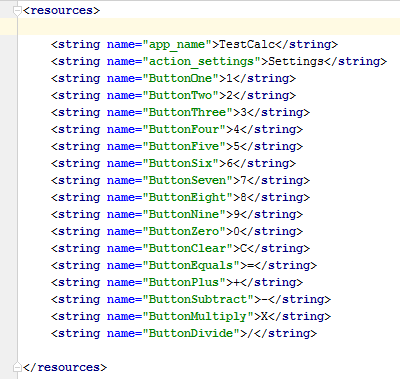
1. Name your first button resource ButtonOne. We are using the ‘full-worded’ name for this resource so we can name the ID of this button the smaller name, Button1. Name all of your button resources similarly.



1. Once you have all of the buttons named, we can check out the string.xml file. Go to App > Res > Values > and double click strings.xml.



1. Inside the file, we will be able to see all of the string values we set, as well as we can change the text within the buttons. See below:



1. Once you have this completed, your buttons will be smaller, fitting to the text inside of them. You will need to adjust the button sizes manually. Navigate back to activity\_main.xml. This file could be found in a tab near the top of the screen, or you can go through App > Res > Layout > and double click activity\_main.xml to reopen the file. Once the file is opened, look at the very bottom of the Palette and you will find two tabs, Design, and Text. Click on Text.
2. For each of the number buttons, and clear button, you will need to add the following line of code:

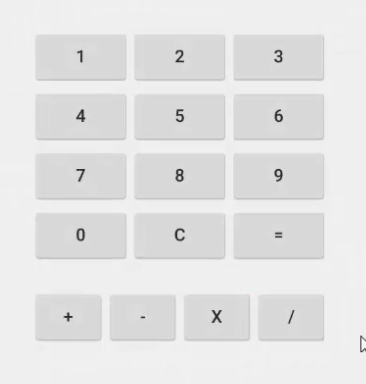
android:minWidth=”80dp”

This will adjust the buttons to 80 device pixels.

1. For each of the four operator buttons, you will need to set the width as well:

Android:minWidth=”60dp”

1. Now that the sizes are adjusted, go back to the design tab and revise your buttons as needed.



1. Next, we need to add another widget, so that we can display our numbers to the screen. From the palette, drag the “Number” text field over to the top area of our calculator and drop it in. Center the widget as needed.
2. Now that we have the design for our application, it’s time to get ready to start coding. Before we can do that, we need to set id’s for our buttons, as well as set an id for our text field. To do so, the quickest way is to double-click each widget, which will open a menu to change the string resource and id. Though you can name these as you wish, I have named mine button1, button2, and etc. Then for the operators: buttonA, buttonS, buttonM, and buttonD. The text field could be named output.
3. Now that our id’s are all set, we need to move on to the programming. Navigate to App > Java, then under the first subfolder you will find MainActivity.java. Double click this to open.
4. First things first, we need to import a few libraries. Add the following just below the other libraries.

*// Additional Imports***import** android.view.View; *// Needed For Widgets***import** android.widget.Button; *// Needed For Buttons***import** android.widget.TextView; *// Needed For Text Field*

1. Within the onCreate method, we need to create objects for each of our buttons so that we can reference them.

*// Buttons 0-9*Button button0 = (Button)findViewById(R.id.***button0***);  
Button button1 = (Button)findViewById(R.id.***button1***);  
Button button2 = (Button)findViewById(R.id.***button2***);  
Button button3 = (Button)findViewById(R.id.***button3***);  
Button button4 = (Button)findViewById(R.id.***button4***);  
Button button5 = (Button)findViewById(R.id.***button5***);  
Button button6 = (Button)findViewById(R.id.***button6***);  
Button button7 = (Button)findViewById(R.id.***button7***);  
Button button8 = (Button)findViewById(R.id.***button8***);  
Button button9 = (Button)findViewById(R.id.***button9***);  
  
*// Buttons Addition, Subtraction, Multiplication, Division*Button buttonA = (Button)findViewById(R.id.***buttonA***);  
Button buttonS = (Button)findViewById(R.id.***buttonS***);  
Button buttonM = (Button)findViewById(R.id.***buttonM***);  
Button buttonD = (Button)findViewById(R.id.***buttonD***);  
  
*// Buttons Clear, Equals*Button buttonC = (Button)findViewById(R.id.***buttonC***);  
Button buttonE = (Button)findViewById(R.id.***buttonE***);

1. Next, we need to create a few variables, which should be done outside, but above, the onCreate Method. The sign variable will hold which sign was used, and the doubles will hold values entered by the user.

**public** String **sign** = **""**;  
**public** Double **tempDouble**, **tempDouble2**;

1. Beneath our buttons, inside the onClick method, we will now start creating the listeners for our buttons. Here I provide the code for button “0,” but you should copy this code, and make needed changes for buttons up to 9.

*// Button 0*button0.setOnClickListener(  
 **new** Button.OnClickListener()  
 {  
 **public void** onClick(View v)  
 {  
 TextView output = (TextView)findViewById(R.id.***outputText***);  
 output.append(**"0"**);  
 }  
 }  
);

1. Once you have completed the previous buttons, you may copy the following code for the rest of the buttons.

*// Button Addition*buttonA.setOnClickListener(  
 **new** Button.OnClickListener()  
 {  
 **public void** onClick(View v)  
 {  
 TextView output = (TextView)findViewById(R.id.***outputText***);  
 **tempDouble** = Double.*parseDouble*(output.getText().toString());  
 output.setText(**""**);  
 **sign** = **"+"**;  
 }  
 }  
);  
  
*// Button Subtraction*buttonS.setOnClickListener(  
 **new** Button.OnClickListener()  
 {  
 **public void** onClick(View v)  
 {  
 TextView output = (TextView)findViewById(R.id.***outputText***);  
 **tempDouble** = Double.*parseDouble*(output.getText().toString());  
 output.setText(**""**);  
 **sign** = **"-"**;  
 }  
 }  
);  
  
*// Button Multiplication*buttonM.setOnClickListener(  
 **new** Button.OnClickListener()  
 {  
 **public void** onClick(View v)  
 {  
 TextView output = (TextView)findViewById(R.id.***outputText***);  
 **tempDouble** = Double.*parseDouble*(output.getText().toString());  
 output.setText(**""**);  
 **sign** = **"\*"**;  
 }  
 }  
);  
  
*// Button Division*buttonD.setOnClickListener(  
 **new** Button.OnClickListener()  
 {  
 **public void** onClick(View v)  
 {  
 TextView output = (TextView)findViewById(R.id.***outputText***);  
 **tempDouble** = Double.*parseDouble*(output.getText().toString());  
 output.setText(**""**);  
 **sign** = **"/"**;  
 }  
 }  
);  
  
*// Button Clear*buttonC.setOnClickListener(  
 **new** Button.OnClickListener()  
 {  
 **public void** onClick(View v)  
 {  
 TextView output = (TextView)findViewById(R.id.***outputText***);  
 output.setText(**""**);  
 }  
 }  
);  
  
*// Button Equals*buttonE.setOnClickListener(  
 **new** Button.OnClickListener()  
 {  
 **public void** onClick(View v)  
 {  
 TextView output = (TextView)findViewById(R.id.***outputText***);  
 **tempDouble2** = Double.*parseDouble*(output.getText().toString());  
  
 **if** (**sign** == **"+"**)  
 {  
 output.setText(Double.*toString*(**tempDouble** + **tempDouble2**));  
 }  
 **else if** (**sign** == **"-"**)  
 {  
 output.setText(Double.*toString*(**tempDouble** - **tempDouble2**));  
 }  
 **else if** (**sign** == **"\*"**)  
 {  
 output.setText(Double.*toString*(**tempDouble** \* **tempDouble2**));  
 }  
 **else if** (**sign** == **"/"**)  
 {  
 **if** (**tempDouble2** == 0)  
 {  
 *// Cannot Divide By Zero* output.setText(**"Cannot Divide By Zero!"**);  
 }  
 **else** {  
 output.setText(Double.*toString*(**tempDouble** / **tempDouble2**));  
 }  
 }  
  
 *// Reset the Sign variable* **sign** = **""**;  
 }  
 }  
);

1. Once saved, you should have a working calculator. Keep in mind that TONS of algorithms go into creating a great calculator. Given that this tutorial was to jumpstart you on how to use Android Studio, making a perfect calculator would be a little… over the top. Save, and run that emulator to see how the program works. For better explanations please see the video provided where I go into detail.

Part 1 Video Link: <https://www.youtube.com/watch?v=sJzLqcBUaKQ&list=PLFVlCGwfyegYi8G0yxIVlGfjT3xGzCZOz&index=9>

Part 2 Video Link: <https://www.youtube.com/watch?v=GfgNkQXcOK8&list=PLFVlCGwfyegYi8G0yxIVlGfjT3xGzCZOz&index=8>

Github Link: <https://github.com/amyork/Android_Studio_Tut_3_Calculator>