

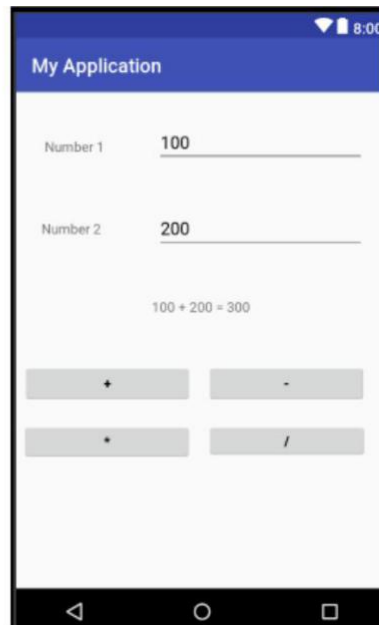
Activities and Intents

1. Create a new android project called "**IntentsProj**".
2. Create two Empty Activities called "**FirstActivity.java**" and "**SecondActivity.java**".
3. Sample interfaces are as follow:

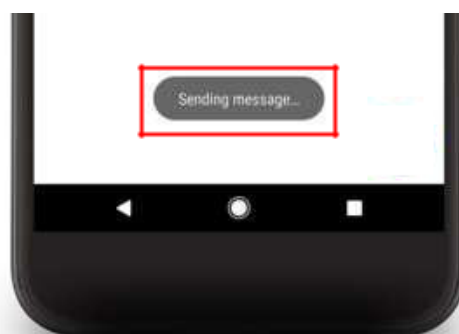
FirstActivity



SecondActivity



4. Implement the button click event of **OK** button to navigate to the "**SecondActivity**".
5. Display a feedback message to the user that **OK** button is just clicked. You can do this by using a **Toast**. A toast provides a simple feedback about an operation in a small popup. It only fills the amount of space required for the message and the current activity remains visible and interactive. Toasts automatically disappear after a timeout.



```
Context context = getApplicationContext(); //The context to
use. Usually your Application or Activity object
CharSequence message = "You just clicked the OK button";
//Display string
int duration = Toast.LENGTH_SHORT; //How long the toast
message will lasts
Toast toast = Toast.makeText(context, message, duration);
toast.show();
```

6. Positioning your Toast. A standard toast notification appears near the bottom of the screen, centered horizontally. You can change this position with the “**setGravity(int, int, int)**” method. This accepts three parameters: a Gravity constant, an x-position offset, and a y-position offset. For example, if you decide that the toast should appear in the top-left corner, you can set the gravity like this:

```
toast.setGravity(Gravity.TOP|Gravity.LEFT, 0, 0);
```

7. Now modify the button click event of OK button to take the input numbers from text fields and send them to the “**SecondActivity**” as Extras.
8. Implement the “**SecondActivity**” to take the extras from “**onCreate**” and display the calculated value in a “**TextView**” according to the selected operator’s button click event.

Exercise: Creating a custom toast

If a simple text message isn't enough, you can create a customized layout for your toast notification

1. Create another xml file inside the layout directory. Here we are having ImageView and TextView in this xml file:

File: customToast.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/custom_toast_layout"
    android:orientation="vertical"
    android:background="#F14E23" >
```

<ImageView

```
android:id="@+id/custom_toast_image"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:contentDescription="Hello world"
android:src="@drawable/customToastImage"/>
```

<TextView

```
android:id="@+id/custom_toast_message"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:contentDescription="To"
android:text="JavaTpoint custom Toast" />
```

</LinearLayout>

2. Now write the code to display the custom toast

File: MainActivity.java

```
public class MainActivity extends AppCompatActivity {

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

      //Creating the LayoutInflater instance
      LayoutInflater li = getLayoutInflater();
      //Getting the View object as defined in the customtoast.xml file
      View layout = li.inflate(R.layout.customtoast, (ViewGroup)
      findViewById(R.id.custom_toast_layout));

      //Creating the Toast object
      Toast toast = new Toast(getApplicationContext());
      toast.setDuration(Toast.LENGTH_SHORT);
      toast.setGravity(Gravity.CENTER_VERTICAL, 0, 0);
      toast.setView(layout); //setting the view of custom toast layout
      toast.show();
    }
}
```