**Learning Outcomes**

1. Activity Lifecycle
2. Multiple Activities
3. Using Intents to pass data to other activities

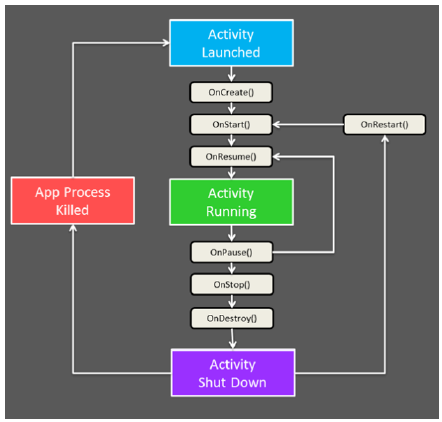
**Activity Lifecycle**

Activity is a programming concept specific for Android development. Activities are building blocks of and Android Application. The Activity can persist in many states. When there is a change in state the appropriate life cycle event is called.

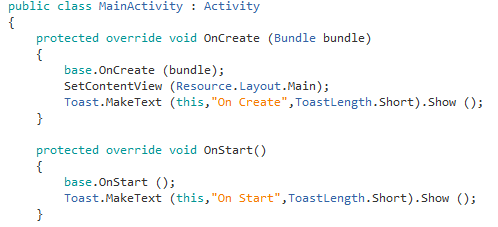
****

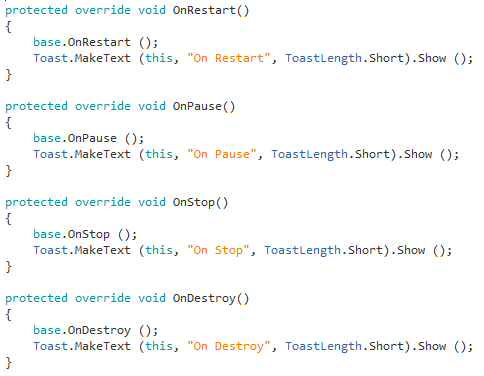
As you see in the above image, an Activity starts when user opens the Activity. An Activity can be in two modes where user can’t see it. The two modes are Pause and Background. In Paused mode the Activity starts from where it left. In Background mode, the activity keeps running but not paused.

Finally the Activity can be stopped and this is the end of Activity.



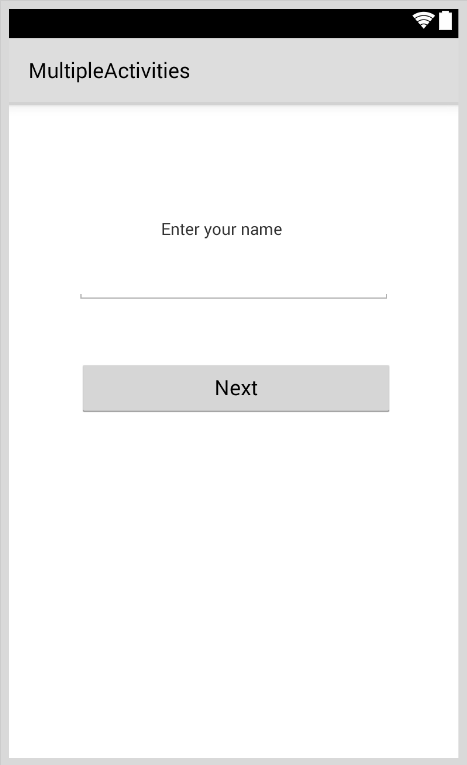
Create an Android application and name it ActivityLifeCycle and write down the following methods.



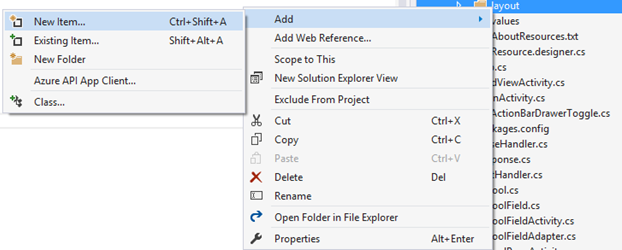


**Multiple Activities**

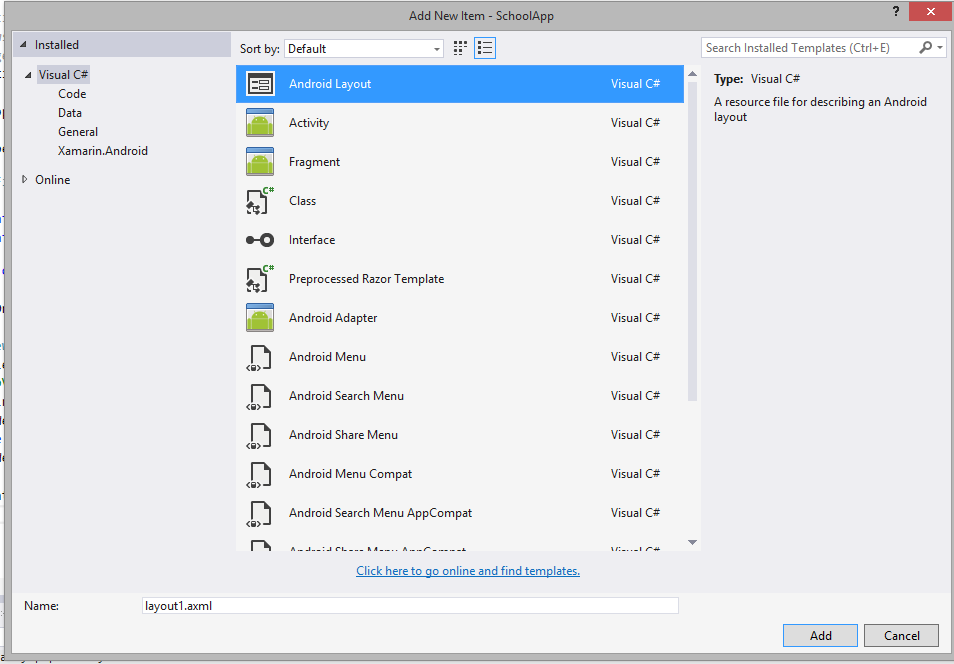
An Android app can contain multiple activities. Design a main activity as shown below.



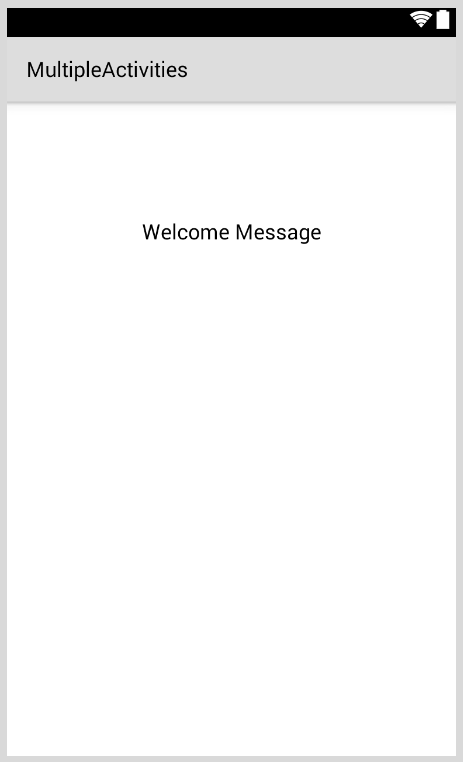
To add another activity, Right click on Layout -> Add -> New Item



Click on layout and give a name

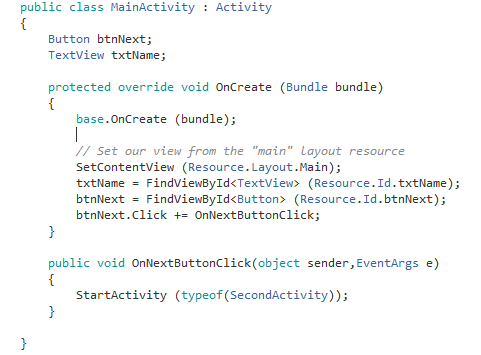


Design the second layout as shown below.

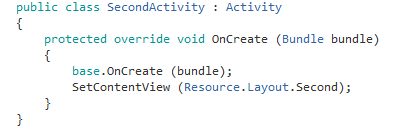


To add the source code file for the layout just created.

**MainActivity.cs**

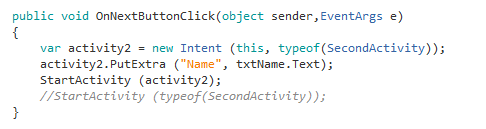


**SecondActivity.cs**

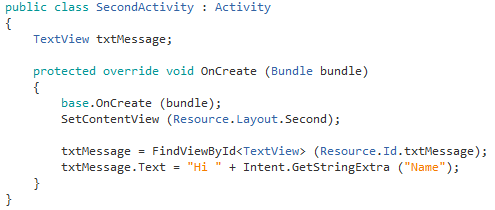


**Passing Data from One Activity to another using Intents**

In **MainActivity.cs** write the following code



**Modify your SecondActivity.cs**

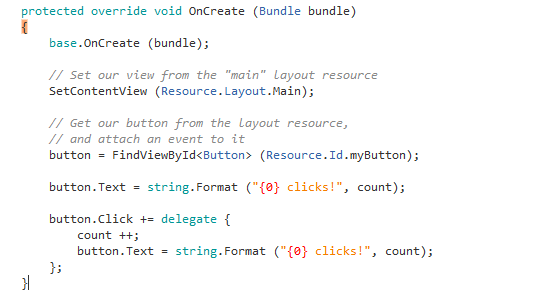


Saving State Changes

When we change the orientation of the mobile app we lose the state of the app.

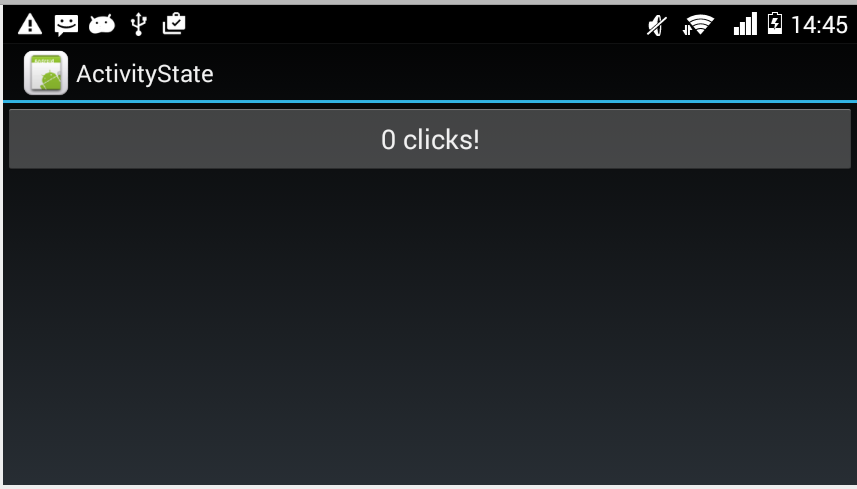
Create a new Android Project

Modify the default code to the following





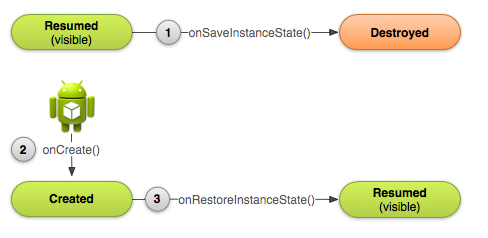
Now when you change the orientation you lose the count



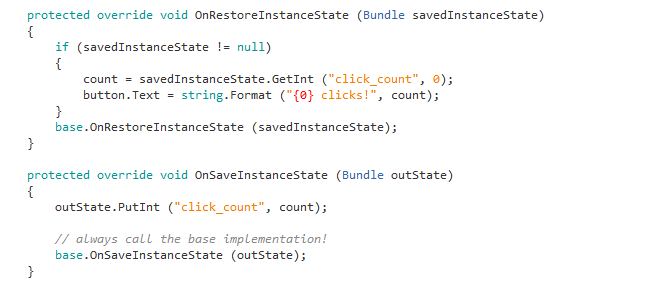
## Adding Code to Preserve Instance State

## Bundle State

The primary option for saving instance state is to use a key/value dictionary object known as a bundle. Recall that when an Activity is created that the OnCreate method is passed a bundle as a parameter, this bundle can be used to restore the instance state.

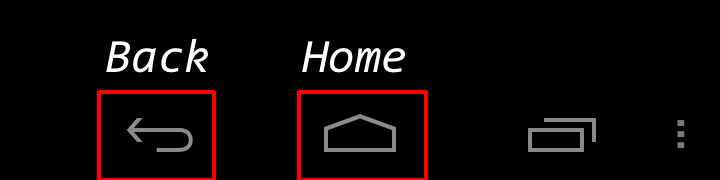


**Add two functions OnSaveInstanceState and OnRestoreInstanceState**



Back vs. Home

Many Android devices have two distinct buttons: a "Back” button and a "Home” button. An example of this can be seen in the following screenshot of Android 4.0.3:

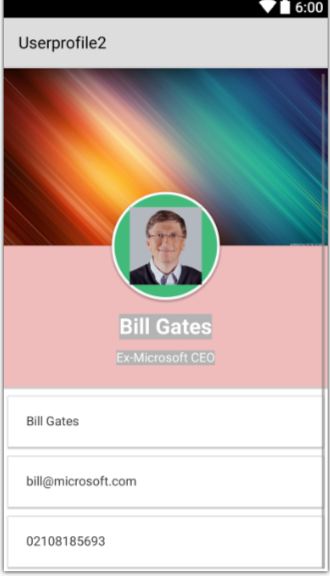
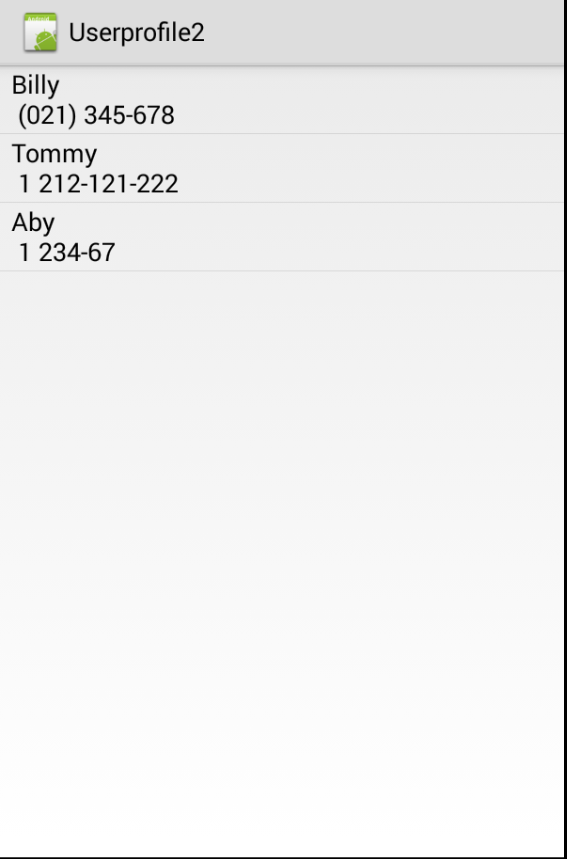
[](http://developer.xamarin.com/guides/android/application_fundamentals/activity_lifecycle/Images/image4.png)

There is a subtle difference between the two buttons, even though they appear to have the same effect of putting an application in the background. When a user clicks the Back button, they are telling Android that they are done with the activity. Android will destroy the Activity. In contrast, when the user clicks the Home button the activity is merely placed into the background – Android will not kill the activity.

**Exercise**

**Userprofile App**

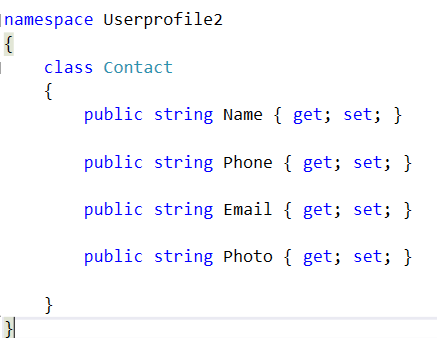
**In this app the first screen will list all the user contacts, when you click on one of the contacts it will open up that contact in detail mode.**



Give it a shot, soluiton on the next page

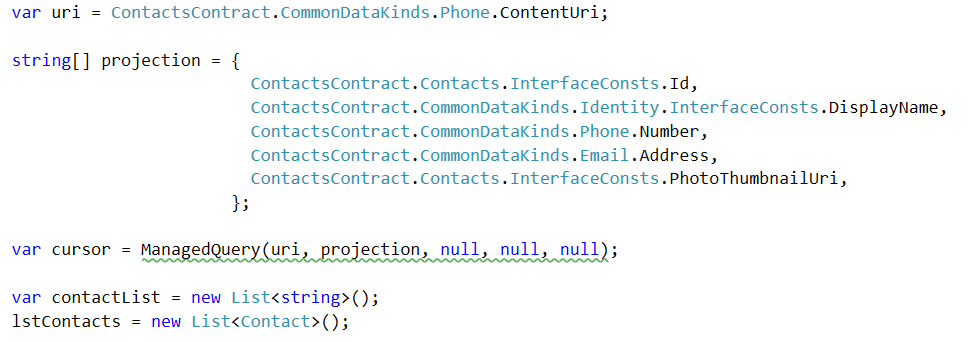
**Solution**

**You will need to create a new class for Contacts**



**MainActivity.cs**









**ProfileActivity.cs**

